

Fast CUR Approximation of Average Matrix and Extensions

Victor Y. Pan^{[1],[a]}, John Svadlenka^{[2],[b]}, and Liang Zhao^{[2],[c]}

^[1] Department of Mathematics and Computer Science
Lehman College of the City University of New York
Bronx, NY 10468 USA

^[2] Ph.D. Programs in Mathematics and Computer Science
The Graduate Center of the City University of New York
New York, NY 10036 USA

^[a] victor.pan@lehman.cuny.edu
<http://comet.lehman.cuny.edu/vpan/>

^[b] jsvadlenka@gradcenter.cuny.edu

^[c] lzhao1@gc.cuny.edu

Abstract

CUR and low-rank approximations are among most fundamental subjects of numerical linear algebra and have applications to a variety of other highly important areas of modern computing as well – they range from machine learning theory and neural networks to data mining and analysis. We first dramatically accelerate computation of such approximations for the average input matrix, then show some narrow classes of hard inputs for our algorithms, and finally narrowing such classes further by means of preprocessing with sparse and structured multipliers. Our extensive numerical tests with a variety of real world inputs from a Singular Matrix Database have consistently produced reasonably close CUR approximations at a low computational cost, and we conclude with novel extensions of our acceleration of CUR and low-rank approximation to the Fast Multipole Method and Conjugate Gradient Algorithms. The techniques of our algorithm design and analysis and some auxiliary results, e.g., on convergence of the probability distribution of the products of randomized bidiagonal matrices to Gaussian distribution, can be of independent interest.

Keywords: CUR approximation, Low-rank approximation, Average input, Gaussian matrices, Cross-approximation, Maximal volume, Fast Multipole Method, Conjugate Gradient Algorithms.

1 Introduction

1.1 CUR and low-rank approximation and our algorithmic goals

Low-rank approximation of an $m \times n$ matrix W having a small numerical rank r , that is, having a well-conditioned rank- r matrix nearby, is one of the most fundamental problems of numerical linear algebra [HMT11] and has a variety of applications to other highly important areas of modern computing as well – they range from machine learning theory and neural

networks [DZBLCF14], [JVZ14], [SKSAR13], [SL14], [SMH07] to numerous problems of data mining and analysis [M11].

One of the most studied approaches to the solution of this problem is given by *CUR approximation* where C and R are a pair of $m \times l$ and $k \times n$ submatrices formed by l columns and k rows of the matrix W , respectively, and U is a $k \times l$ matrix such that $W \approx CUR$ and $\text{rank}(U)$ is equal to r or as close to r as possible. Every low-rank approximation (including CUR approximations) allows very fast approximate multiplication of the matrix W by a vector, but CUR approximation is particularly transparent and memory efficient.

Seeking these approximations one tries to decrease

- (i) the complexity of their computation and
- (ii) the error norms of the approximation.

We seek algorithms that use $O((m+n)r^2)$ flops.¹ For $r^2 \ll \min\{m, n\}$, this is much less than the information lower bound $mn/2$ (which follows because a flop involves at most two entries).

1.2 State of the art and our progress

The algorithms of [GE96] and [P00] compute CUR approximations by using order of $mn \min\{m, n\}$ flops. [BW14] do this in $O(mn \log(mn))$ flops by using randomization.

These are record upper bounds for computing a CUR approximation to *any input matrix* W , but the user could be happy with close CUR approximations to *many matrices* W if they cover the class of his/her interest, and for such restricted input classes we need much fewer than $mn/2$ flops.

Let us further discuss our progress. We devise two-stage algorithms:

- (i) we first fix a pair of integers $k \leq m$ and $l \leq n$ and, by using the algorithms of [GE96] or [P00], compute a CUR approximation to a random $k \times l$ submatrix;
- (ii) then we compute a CUR approximation of an input matrix W itself.

Our algorithms fail for a specific class of input matrices W of our Example 11, but such a class of hard inputs is narrow because the algorithms produce close CUR approximations to the average input matrix over all $m \times n$ input matrices W of numerical rank r , where the average is under the standard Gaussian (normal) probability distribution for the $(m+n)r$ i.i.d. input parameters.²

By extending our two-stage algorithms with the technique of [GOSTZ10], which we call *cross-approximation*, we a little narrow the class of hard inputs of Example 11 to the smaller class of Example 14. Moreover for this approach we deduce a sharper bounds on the error of CUR approximation, and we prove that *this is achieved already in a single two-stage loop* of cross-approximation. In this case we advance the known direction of maximizing the *volume of the submatrix* that defines a CUR approximation, but we also exploit additional advantages of choosing proper rectangular submatrix and optimize its size towards maximizing its *projective volume*.

For alternative and complementary technique, we pre-process an input matrix W with appropriate multipliers and then apply our algorithms to the products. We prove that our algorithms are likely to output accurate CUR approximations of the product of any matrix having numerical rank r with a Gaussian multiplier.³ The result is a low-rank approximation

¹Here and hereafter “*flop*” stands for “floating point arithmetic operation”.

²Here and hereafter the acronym “*i.i.d.*” stands for “independent identically distributed”, and we refer to “standard Gaussian random” variables and matrices just as “*Gaussian*”.

³Here and hereafter “*likely*” means “with a probability close to 1”.

of the original input matrix, being not its CUR approximation, but a minor generalization.

Computing the product of W with such multipliers is expensive, however, and we replace them by sparse and structured multipliers, for which the computation of the products is not costly; then again we seek algorithms that succeed in CUR approximation of many inputs (rather than of any input). We prove that indeed our algorithms succeed for the average input and for any square unitary multiplier. Moreover, the narrow classes of bad inputs vary when we vary multipliers, and the intersection of such classes shrinks.

The paper [PZ16] studied a similar preprocessing for a distinct purpose and proposed efficient preprocessing policies and some families of sparse and structured unitary multipliers, for which preprocessing is simple; we adopt all this for our inputs as well.

In Section 6.4 we prove that probability distributions of the products of random and randomly permuted bidiagonal matrices converge to Gaussian as the number of factors grows to infinity. (Our non-trivial proof occupies more than 3 pages.) The result has independent interest, and in our tests convergence was consistently quite fast, although the preprocessing was still expensive. The convergence proof and test result, however, may motivate further effort in this direction.

Converse direction seems to be practically much more promising. Namely, instead of converging to an $n \times n$ Gaussian multiplier, we begin with it, then sparsify it directly or in two steps, that is, by decomposing it into the product of $2n - 2$ bidiagonal factors and then dropping most of them. In both sparsifications the goal is the decrease of the overall complexity. There is a variety of promising modifications, and their choice is a challenge for our next empirical study.

In our extensive tests with a variety of real world input data for matrices from a Singular Matrix Database, our fast algorithms have already consistently produced close CUR approximations. We have also provided evidence of how various pre-processing multipliers can help with bad inputs.

Finally, our algorithms can be extended to the acceleration of various computational problems that are known to have links to low-rank approximation, and in our concluding Section 8 we describe a novel and rather unexpected extensions to the acceleration of the Fast Multipole Method and Conjugate Gradient Algorithms,⁴ both being among the most celebrated achievements of the 20th century in Numerical Linear Algebra (e.g., see some comments on the FMM in [C00] and [BY13] and on the CG algorithms in [GL13, Section 11.3]).

1.3 Some related results on matrix algorithms and our progress on other fundamental subjects of matrix computations

A huge bibliography on CUR and low-rank approximation, including the known best algorithms, which we already cited, can be accessed from the papers [HMT11], [M11], [BW14] and [W14].

Our main contribution is dramatic acceleration of the known algorithms.

Some of our techniques extend the ones of [PZ16], [PZ17], and [PZa], where we also show duality of randomization and de-randomization and apply it to fundamental matrix computations.

In [PZ16] we first prove that preprocessing with almost any well-conditioned multiplier of full rank is as efficient on the average for low-rank approximation as preprocessing with

⁴Hereafter we use the acronyms FMM and CG.

a Gaussian one, and then we propose some new highly efficient sparse and structured multipliers. Besides providing a new insight into the subject, this motivates the design of more efficient algorithms and shows specific direction to this goal.

We obtain similar progress in [PZa] for preprocessing Gaussian elimination with no pivoting and block Gaussian elimination. We recall that Gaussian elimination with partial pivoting is performed millions time per day, where pivoting, required for numerical stabilization, is frequently a bottleneck because it interrupts the stream of arithmetic operations with foreign operations of comparison, involves book-keeping, compromises data locality, and increases communication overhead and data dependence.

Randomized preprocessing is a natural substitution for pivoting, and in [PZa] we show that Gaussian elimination with no pivoting as well as block Gaussian elimination (which is another valuable algorithm and which also requires protection against numerical problems) are efficient on the average input with preprocessing by any non-singular and well-conditioned multipliers.

[PZ17] obtains similar progress for the important subject of the approximation of trailing singular spaces associated with the ν smallest singular values of a matrix having numerical nullity ν .

Our current progress greatly supersedes these earlier results, however, in terms of the scale of the acceleration of the known algorithms.

Our study of CUR approximation for the average input (including the approach itself, proposed techniques and formal and empirical results of our analysis), our technique of representing random Gaussian multipliers as a product of random bidiagonal factors, the proof of convergence of the probability distribution of the product to Gaussian distribution, and our extension of CUR approximation to FMM and CG algorithms are new and can be of some independent interest.

1.4 Organization of the paper

In the next subsection and in Appendix A we recall some basic definitions. In Section 2 we define CUR approximation via two index sets and a basic matrix and express approximation errors via the norm of the basic matrix. In Section 3 we propose and analyze a fast algorithm for a CUR approximation of the average matrix and show a narrow class of hard inputs for which the algorithm fails. In Section 4 we describe generic cross-approximation algorithm, which is efficient for a more general class of inputs, but fails for a more narrow input class. In Section 5 we analyse this algorithm by using maximization of the volume of a submatrix of the input matrix and strengthen our bounds on the output errors in the case of the average input, and consequently to a large class of specific inputs. In Section 6 we study extension of our algorithms to a larger input class by applying sparse and structured preprocessing with unitary multipliers. In Section 6 we cover preprocessing with (i) Gaussian and (ii) sparse and structured multipliers, prove that they support computation of a CUR approximation of any matrix (with high probability) in case (i) and definitely for the average matrix in case (ii), and prove a non-trivial result on randomized bidiagonal factorization of Gaussian matrices. In Section 7 we describe the results of our extensive numerical tests. In Section 8 we briefly summarize our progress and then describe its two novel extensions. In Appendix B we recall some known algorithms and estimates for volume maximization.

1.5 Basic definitions

The concepts “large”, “small”, “near”, “close”, “approximate”, “ill-conditioned” and “well-conditioned” are quantified either in context or explicitly, if needed.

“ \gg ” and “ \ll ” mean “much greater than” and “much less than”, respectively.

The acronym “*w.l.o.g.*” stands for “without loss of generality”.

$|\mathcal{S}|$ denotes the cardinality of a set \mathcal{S} .

$\mathbb{C}^{m \times n}$ is the class of $m \times n$ matrices with complex entries.

I_s denotes the $s \times s$ identity matrix.

$\text{diag}(B_1, \dots, B_k) = \text{diag}(B_j)_{j=1}^k$ is a $k \times k$ block diagonal matrix with diagonal blocks B_1, \dots, B_k .

$(B_1 \mid \dots \mid B_k)$ and (B_1, \dots, B_k) denote a $1 \times k$ block matrix with blocks B_1, \dots, B_k .

W^T and W^* denote the transpose and the Hermitian transpose of an $m \times n$ matrix $W = (w_{ij})_{i,j=1}^{m,n}$, respectively. ($W^* = W^T$ if the matrix W is real.)

$\|W\|_h$, for $h = 1, 2, \infty$, $\|W\| = \|W\|_2$, $\|W\|_F$, and $\|W\|_C$ denote its h -norm, spectral, Frobenius and Chebyshev norms, respectively, such that (cf. [GL13, Section 2.3.2 and Corollary 2.3.2]),

$$\begin{aligned} \|W\|_h &:= \sup_{\|\mathbf{v}\|_h=1} \|W\mathbf{v}\|_h, \quad \|W\|_F^2 := \sum_{i,j=1}^{m,n} |w_{ij}|^2 = \sum_{j=1}^{\text{rank}(W)} \sigma_j^2(W), \quad \|W\|_C := \max_{i,j=1}^{m,n} |w_{ij}|, \\ \|W\|_1 &= \|W^*\|_\infty = \max_{j=1,\dots,n} \sum_{i=1}^m |w_{ij}|, \quad \frac{1}{\sqrt{m}} \|W\|_1 \leq \|W\| \leq \sqrt{n} \|W\|_1, \quad \|W\|^2 \leq \|W\|_1 \|W\|_\infty, \\ \|W\|_C &\leq \|W\| \leq \sqrt{mn} \|W\|_C, \quad \|W\|^2 \leq \|W\|_F^2 \leq \min\{m, n\} \|W\|^2. \end{aligned} \tag{1}$$

Wherever this causes no confusion we unify our study by writing $|\cdot|$ for both spectral and Frobenius matrix norm.

An $m \times n$ matrix W is *unitary* (also called *orthogonal* when it is real) if $W^*W = I_n$ or $WW^* = I_m$.

$W := S_W \Sigma_W T_W^*$ denotes its compact SVD, such that $S_W^* S_W = S_W S_W^* = I_m$, $T_W^* T_W = T_W T_W^* = I_n$, $\Sigma_W := \text{diag}(\sigma_j(W))_{j=1}^{\text{rank}(W)}$, $\sigma_j(W)$ denotes the j th largest singular value of W for $j = 1, \dots, \text{rank}(W)$, and $\sigma_j(W) = 0$ for $j > \text{rank}(W)$.

U is its left inverse if $UW = I_n$ and its right inverse if $WU = I_m$.

$W^+ := T_W \Sigma_W^{-1} S_W^*$ is its Moore–Penrose pseudo-inverse. W^+ is a left inverse of W if $\text{rank}(W) = n$ and is a right inverse of W if $\text{rank}(W) = m$.

$\kappa(W) := \sigma_1(W)/\sigma_\rho(W) = |W| |W^+| \geq 1$ denotes the *condition number* of W .

A matrix W is unitary if and only if $\kappa(W) = 1$, is *ill-conditioned* if $\kappa(W)$ is large in context, and is *well-conditioned* if $\kappa(W)$ is reasonably bounded.

A matrix W has *numerical rank* r (and then we write $\text{nrank}(W) = r$) if a nearby well-conditioned matrix has rank r . (A matrix is ill-conditioned if and only if it has a matrix of a smaller rank nearby or equivalently if and only if its rank exceeds its numerical rank.)

We recall some definitions and basic results for random matrices in Appendix A and state some other definitions just before we use them.

2 CUR Approximation: Definition and Error Estimates

2.1 The basic expression

Definition 1. (Cf. [GZT95], [GTZ97], [GTZ97a], [BW14].) For a matrix $W = (w_{i,j})_{i,j=1}^{m,n} \in \mathbb{C}^{m \times n}$ and two sets $\mathcal{I} \subseteq \{1, \dots, m\}$ of cardinality $k \leq m$ and $\mathcal{J} \subseteq \{1, \dots, n\}$ of cardinality $l \leq n$, define the submatrices $W_{\mathcal{I},:} := (w_{i,j})_{i \in \mathcal{I}; j=1, \dots, n}$, $W_{:, \mathcal{J}} := (w_{i,j})_{i=1, \dots, m; j \in \mathcal{J}}$, and $W_{\mathcal{I}, \mathcal{J}} := (w_{i,j})_{i \in \mathcal{I}; j \in \mathcal{J}}$. Then these two sets and a $k \times l$ basic matrix U of rank $O(r)$ define a *CUR approximation* of the matrix W ,

$$\bar{W} := CUR \text{ for } C = W_{:, \mathcal{J}} \text{ and } R = W_{\mathcal{I},:}, \quad (3)$$

if $\bar{W} \approx W$. This is a *CUR decomposition* if $\bar{W} = W$.

Theorem 2. Let $W_{\mathcal{I}, \mathcal{J}}$ be a $k \times l$ submatrix of an $m \times n$ matrix W , let $r := \text{rank}(W_{\mathcal{I}, \mathcal{J}}) = \min\{k, l\}$ and let U be a left or right inverse of the matrix $W_{\mathcal{I}, \mathcal{J}}$, e.g.,

$$U = W_{\mathcal{I}, \mathcal{J}}^+. \quad (4)$$

Then $W = \bar{W}$ for the matrix \bar{W} of (3) if and only if $\text{rank}(W) = r$.

Proof. The “only if” claim is obvious. Let us prove the “if” claim.

W.l.o.g. suppose that $r = k \leq l$ and $W_{\mathcal{I}, \mathcal{J}}U = I_r$. Then $W_{:, \mathcal{J}}U = (I_r \mid X)^*$ for a $r \times (m - r)$ matrix X , and so

$$\bar{W} = W_{:, \mathcal{J}}UW_{\mathcal{I},:} = (I_r \mid X)^*W_{\mathcal{I},:} = \begin{pmatrix} W_{\mathcal{I},:} \\ Y \end{pmatrix},$$

for a $(m - k) \times k$ matrix Y , that is, the matrices W and \bar{W} share their first r rows.

Next, w.l.o.g. let the matrix $W_{\mathcal{I}, \mathcal{I}}$ be non-singular. (Otherwise we could have achieved such a nonsingularity by interchanging the rows of $W_{:, \mathcal{J}}$ and the columns of $W_{\mathcal{I},:}$.)

Write $\bar{W} = MN$ where $M = (I_r \mid X)^*W_{\mathcal{I}, \mathcal{I}} = \begin{pmatrix} W_{\mathcal{I}, \mathcal{I}} \\ T \end{pmatrix}$, $N = W_{\mathcal{I}, \mathcal{I}}^{-1}W_{\mathcal{I},:} = (I_r \mid Z)$, T is an $(m - r) \times r$ matrix, and Z is a $r \times (n - r)$ matrix. Conclude that the rank- r matrices W and $\bar{W} = MN$ share their first r columns as well, and the theorem follows. \square

2.2 Basic error estimates

Theorem 2 implies that equations (3) and (4) define CUR decomposition for any matrix $W = \bar{W}$ of rank r if the submatrix $W_{\mathcal{I}, \mathcal{J}}$ has full rank r . Now let W' denote the closest rank- r approximation of W , obtained by setting to 0 all singular values $\sigma_j(W)$ for $j > r$. Then write

$$\sigma'_{r+1}(W) := |W - W'|, \quad (5)$$

$\sigma'_{r+1}(W) = \sigma_{r+1}(W)$ for $|\cdot| = \|\cdot\|$ and $(\sigma'_{r+1}(W))^2 = \sum_{j=r+1}^{\text{rank}(W)} \sigma_j^2(W)$ for $|\cdot| = \|\cdot\|_F$ and let

$$W' = C'U'R' \quad (6)$$

denote a CUR decomposition. Then we define the approximation error matrices as follows,

$$\Delta := W - \bar{W}, \quad \Delta' := W' - \bar{W} = C'U'R' - CUR, \text{ and } |\Delta| \leq |\Delta'| + \sigma'_{r+1}(W). \quad (7)$$

Theorem 3. Let $|\cdot|$ denote a spectral or Frobenius norm. Define two CUR decompositions $\bar{W} = CUR$ by (3) and $W' = C'U'R'$ by (6) and write

$$\epsilon_C := |C' - C|/|C|, \quad \epsilon_U := |U' - U|/|U| \quad \text{and} \quad \epsilon_R := |R' - R|/|R|. \quad (8)$$

Then

$$|C'U'R' - CUR| \leq \theta |C| |U| |R|, \quad \text{for } \theta = (1 + \epsilon_C)(1 + \epsilon_U)(1 + \epsilon_R) - 1. \quad (9)$$

Proof. Notice that

$$C'U'R' - CUR = (C' - C)U'R' + CU'(R' - R) + C(U' - U)R.$$

Therefore

$$|C'U'R' - CUR| \leq |C' - C| |U'| |R'| + |C| |U'| |R' - R| + |C| |U' - U| |R|.$$

Substitute (8) and obtain (9). \square

Corollary 4. Let the assumptions of Theorem 3 hold for $C = W_{:, \mathcal{J}}$, $R = W_{\mathcal{I}, :}$, $U = W_{\mathcal{I}, \mathcal{J}}^+$, and $U' = (W'_{\mathcal{I}, \mathcal{J}})^+$ and write

$$\epsilon_E := \frac{|W'_{\mathcal{I}, \mathcal{J}} - W_{\mathcal{I}, \mathcal{J}}|}{|W_{\mathcal{I}, \mathcal{J}}|}, \quad \epsilon_{\mathcal{I}, \mathcal{J}} = \epsilon_E \kappa(W_{\mathcal{I}, \mathcal{J}}), \quad \kappa(W_{\mathcal{I}, \mathcal{J}}) = |W_{\mathcal{I}, \mathcal{J}}| |U|, \quad \text{and } \epsilon := \min\{\epsilon_C, \epsilon_R, \epsilon_E\}. \quad (10)$$

(i) Let $|\cdot| = \|\cdot\|$ and $\epsilon_{\mathcal{I}, \mathcal{J}} < 1$, define ϵ_C , ϵ_R , ϵ_U , and ϵ_E under the spectral norm, and let the matrices $W_{\mathcal{I}, \mathcal{J}}$ and $W'_{\mathcal{I}, \mathcal{J}}$ be nonsingular (cf. [S98, Corollary 1.4.19]). Then bound (9) holds for

$$\theta = \theta(\epsilon, |U|) \leq (1 + \epsilon)^2 \left(1 + \frac{\epsilon_{\mathcal{I}, \mathcal{J}}}{1 - \epsilon_{\mathcal{I}, \mathcal{J}}}\right) - 1 = (2 + \kappa(W_{\mathcal{I}, \mathcal{J}}))(1 + O(\epsilon|U|))\epsilon. \quad (11)$$

(ii) Let $|\cdot| = \|\cdot\|_F$ and define ϵ_C , ϵ_R , ϵ_U , and ϵ_E under the Frobenius norm. Then (9) holds for

$$\theta \leq (1 + \epsilon)^2(1 + 2\epsilon u) - 1 = (1 + u)(2 + O(\epsilon)) \epsilon, \quad u = \frac{|W_{\mathcal{I}, \mathcal{J}}|}{|U|} \max\{\|U\|^2, \|U'\|^2\}. \quad (12)$$

Proof. Combine (9) and (10) with the bounds $\epsilon_U \leq \frac{\epsilon_{\mathcal{I}, \mathcal{J}}}{1 - \epsilon_{\mathcal{I}, \mathcal{J}}}$ and $\epsilon_U |U| \leq 2\epsilon_E \frac{|W_{\mathcal{I}, \mathcal{J}}|}{|U|} \max\{|U|^2, |U'|^2\}$ of [S98, Corollary 1.4.19] and [GL13, Section 5.5.3] and obtain (11) and (12), respectively. \square

Remark 5. Recall that $\epsilon = \min\{\epsilon_C |C|, \epsilon_E |W_{\mathcal{I}, \mathcal{J}}|, \epsilon_R |R|\} \leq \sigma'_{r+1}(W)$ and that $|W_{\mathcal{I}, \mathcal{J}}| \leq \min\{|C|, |R|\}$ because $C = W_{:, \mathcal{J}}$ and $R = W_{\mathcal{I}, :}$. Therefore $\epsilon \leq \epsilon_E \leq \sigma'_{r+1}(W) / \max\{|C|, |R|\}$.

3 Fast CUR Approximation of the Average Matrix

3.1 Some known algorithms for CUR approximation

Suppose that an $m \times n$ input matrix W has numerical rank r , that is, the ratio $\|W\|/\sigma_r(W)$ is not large, while $\sigma_r(W) \gg \sigma_{r+1}(W)$. Then, by virtue of Corollary 4, equations (3) and

(4) define a close CUR approximation \tilde{W} to the matrix W if the submatrix $W_{\mathcal{I},\mathcal{J}}$ has rank r and if

$$\sigma_r(W_{\mathcal{I},\mathcal{J}}) \geq \sigma_r(W)/p \quad (13)$$

where the value $1/p$ is not small, say, $p = p(m, n, r)$ is a low degree polynomial in m , n , and r .

By using $O((mn \min\{m, n\})$ flops [GE96, Algorithm 4 for $f = 1$] and [P00, Algorithm 3 for $\mu = 1$]⁵ compute strong rank-revealing QR and LU factorizations, respectively, defining submatrices $W_{\mathcal{I},\mathcal{J}}$ that satisfy (13) for $p = \sqrt{(u-r)r+1}$ and $p = (u-r)r+1$, respectively, where $u = \max\{m, n\}$.

Randomized algorithms of [BW14] use $O((mn \log(n) + (m+n) \text{poly}(r, \log(n), 1/\epsilon)))$ flops and with a constant probability of success output matrices $C = W_{:, \mathcal{J}} \in \mathbb{C}^{m \times l}$, $R = W_{\mathcal{I}, :} \in \mathbb{C}^{k \times n}$, and $U \in \mathbb{C}^{k \times l}$ such that $\text{rank}(U) = r$, $k = O(r/\epsilon)$, $l = O(r/\epsilon)$, and the Frobenius norm of the error matrix $\Delta = W - CUR$ is optimal up to a factor of $1 + \epsilon$.

3.2 A fast algorithm for a basic matrix

Next we specify and analyze a simple fast algorithm that computes a basic matrix defining a CUR approximation of the average $m \times n$ matrix W of numerical rank r . It uses $O((m+n)r^2)$ flops, that is, $o(mn)$ if $r^2 = o(\min\{m, n\})$.

Algorithm 6. *Fast computation of a basic matrix.*

INPUT: Three integers m , n , and r such that $1 \leq r \leq \min\{m, n\}$ and an $m \times n$ matrix W .

INITIALIZATION: Fix two integers k and l such that $r \leq k \leq m$, $r \leq l \leq n$ (see Remark 9).

OUTPUT: Two sets \mathcal{I} and \mathcal{J} of row and column indices defining a $r \times r$ submatrix $W_{\mathcal{I},\mathcal{J}}$ of W and its inverse $U := W_{\mathcal{I},\mathcal{J}}^{-1}$. [They define a CUR approximation of the matrix W , for $C = W_{:, \mathcal{J}}$, $U = W_{\mathcal{I}, \mathcal{J}}^{-1}$ and $R = W_{\mathcal{I}, :}$]⁶

COMPUTATIONS: 1. Choose two basic sets of integer indices,

$$\mathcal{I}' = \{i_1, \dots, i_k\} \text{ and } \mathcal{J}' = \{j_1, \dots, j_l\},$$

for $1 \leq i_1 < \dots < i_k \leq m$ and $1 \leq j_1 < \dots < j_l \leq n$.

2. Apply [GE96, Algorithm 4 for $f = 1$] or [P00, Algorithm 1 for $\mu = 1$] for $m = k$, $n = l$, $W = W_{\mathcal{I}', \mathcal{J}'}$. Output the inverse $U := W_{\mathcal{I}', \mathcal{J}'}^{-1}$ of the resulting $r \times r$ submatrix $W_{\mathcal{I}, \mathcal{J}}$ of the matrix $W_{\mathcal{I}', \mathcal{J}'}$ and the sets \mathcal{I} and \mathcal{J} .

Complexity of performing the algorithm. stage 1 involves no flops. stage 2 involves $O(kl \min\{k, l\})$ flops, that is, $O(r^3)$ if $k + l = O(r)$, $O(ml^2)$ if $k = m \geq l$, or $O(nk^2)$ if $l = n \geq k$, versus order of $mn \min\{m, n\}$ flops in [GE96] and [P00] and $O(mn \log(n))$ flops in [BW14] if the algorithms of these papers are applied to the matrix W .

Remark 7. By applying randomized algorithms of [BW14] instead of deterministic ones of [GE96] and [P00], we would perform stage 2 by using $O(kl \log(l) + (k+l) \text{poly}(r, 1/\epsilon))$ flops, would yield the Frobenius output error norm $(1+\epsilon)\sigma'_{r+1}$ for any positive ϵ , and would increase the size of the basic matrix to $q \times q$ for q of order r/ϵ .

⁵By using a parameter μ that a little exceeds 1 the paper [P00] controls the impact of rounding errors. We simplify our estimates by assuming that $\mu = 1$, but we can readily extend them to the case where $\mu \geq 1$.

⁶We assume that the computed matrix $W_{\mathcal{I}, \mathcal{J}}$ is nonsingular; otherwise we should either re-apply the algorithm for a distinct pair of the initial sets \mathcal{I} and \mathcal{J} or stop and output FAILURE.

3.3 The spectral norm of the basic matrix in the case of a Gaussian input

We begin with the following lemma.

Lemma 8. *Let $G \in \mathcal{G}^{k \times r}$, $\Sigma \in \mathbb{C}^{r \times r}$, $H \in \mathcal{G}^{r \times l}$, and $\text{rank}(G) = \text{rank}(\Sigma) \text{rank}(H) = r$. Then $\sigma_r(G\Sigma H) = \sigma_r(G)\sigma_r(\Sigma)\sigma_r(H)$.*

Proof. Let $G = S_G \Sigma_G T_G$ and $H = S_H \Sigma_H T_H$ be SVDs where S_G , T_G , D_H , and T_H are unitary matrices, Σ_G and Σ_H are the $r \times r$ nonsingular diagonal matrices of singular values of the matrices G and H , respectively, and T_G and S_H are $r \times r$ unitary matrices.

Write $M := \Sigma_G T_G \Sigma S_H \Sigma_H$. Then $M^{-1} = \Sigma_H^{-1} S_H^* \Sigma^{-1} T_G^* \Sigma_G^{-1}$, and consequently we obtain that $\sigma_r(M) = \sigma_r(G)\sigma_r(\Sigma)\sigma_r(H)$.

Let $M = S_M \Sigma_M T_M$ be SVD where S_M and T_M are $r \times r$ unitary matrices.

Then $S := S_G S_M$ and $T := T_M T_H$ are unitary matrices, and so $G\Sigma H = S\Sigma_M T$ is SVD.

Therefore $\sigma_r(G\Sigma H) = \sigma_r(M) = \sigma_r(G)\sigma_r(\Sigma)\sigma_r(H)$. \square

Next we estimate the norm $\|W_{\mathcal{I}', \mathcal{J}'}^+\|$ in the case where the matrix W is scaled factor-Gaussian with expected rank r ; factor-Gaussian matrices are a special case where $\eta_+ = 1$ (cf. Definition 43).

In this case $W_{\mathcal{I}', \mathcal{J}'} = G\Sigma H$, for $G \in \mathcal{G}^{k \times r}$, $\Sigma \in \mathbb{C}^{r \times r}$, $\eta_+ := \|\Sigma^+\|$, and $H \in \mathcal{G}^{r \times l}$.

Apply Theorem 49 and obtain

$$\text{Probability}\{\|G_{\mathcal{I}', :}^+\| \geq x\} \leq \frac{(x/k)^{-0.5(k-r+1)}}{\Gamma(k-r+2)}; \text{Probability}\{\|H_{:, \mathcal{J}'}^+\| \geq x\} \leq \frac{(x/k)^{-0.5(l-r+1)}}{\Gamma(l-r+2)},$$

$$\mathbb{E}(\|G_{\mathcal{I}', :}^+\|) = \mathbb{E}(\nu_{k,r}^+) \leq \frac{e\sqrt{k}}{k-r}, \text{ and } \mathbb{E}(\|H_{:, \mathcal{J}'}^+\|) = \mathbb{E}(\nu_{r,l}^+) \leq \frac{e\sqrt{l}}{l-r}, \text{ for } e = 2.71828 \dots$$

Lemma 8 implies that

$$\mathbb{E}(\|W_{\mathcal{I}', \mathcal{J}'}^+\|) = \mathbb{E}(\|G_{\mathcal{I}', :}^+\| \|H_{:, \mathcal{J}'}^+\|) \eta_+ = \mathbb{E}(\|G_{\mathcal{I}', :}^+\|) \mathbb{E}(\|H_{:, \mathcal{J}'}^+\|) \eta_+,$$

and so

$$\mathbb{E}(\|W_{\mathcal{I}', \mathcal{J}'}^+\|) \leq \frac{e^2 \sqrt{kl} \eta_+}{(k-r)(l-r)}, \quad e^2 < 7.4. \quad (14)$$

Therefore $\mathbb{E}(\|W_{\mathcal{I}', \mathcal{J}'}^+\|)$ is inversely proportional to k if $k = l \gg r$.

Furthermore the above estimates also imply that $\text{Probability}\{\|W_{\mathcal{I}', \mathcal{J}'}^+\| > \zeta \mathbb{E}(\|W_{\mathcal{I}', \mathcal{J}'}^+\|)\}$ for any fixed $\zeta > 1$ converges to 0 exponentially fast as $\min\{k-r, l-r\}$ grows to infinity.

Next we apply Algorithm 6 to (i) l-factor-Gaussian matrices W choose $k = m$ and (ii) r-factor-Gaussian matrices W choosing $l = n$ (cf. Definition 43) and prove that similar estimates hold under the following *left/right orientation rule*:

(i) choose $k = m$ where W is an l-factor-Gaussian matrix and

(ii) choose $l = n$ where W is a r-factor-Gaussian matrix.

In case (i) an $r \times n$ matrix H is not Gaussian anymore but is well-conditioned with $\|H^+\| = h_+$, and $\mathcal{J}' = \{1, \dots, m\}$. Apply Lemma 8 and deduce that

$$\mathbb{E}(\|W_{\mathcal{I}', \mathcal{J}'}^+\|) = \mathbb{E}(\|G_{\mathcal{I}', :}^+\| \|H_{:, \mathcal{J}'}^+\|) = \mathbb{E}(\|G_{\mathcal{I}', :}^+\|) h_+ = \mathbb{E}(\nu_{k,r}^+) h_+ \leq \frac{e\sqrt{k} \eta_+}{k-r}, \quad e = 2.71828 \dots \quad (15)$$

In case (ii) an $m \times l$ matrix G is not Gaussian anymore but is well-conditioned with $\|G\|^+ = g_+$, and $\mathcal{I}' = \{1, \dots, n\}$. Apply Lemma 8 and deduce that

$$\mathbb{E}(\|W_{\mathcal{I}', \mathcal{J}'}^+\|) = \mathbb{E}(\|G_{\mathcal{I}', :}^+\| \|H_{:, \mathcal{J}'}^+\|) = \mathbb{E}(\|H_{:, \mathcal{J}'}^+\|)g_+ = \mathbb{E}(\nu_{l, r}^+)g_+ \leq \frac{e\sqrt{l}}{l-r}g_+, \quad e = 2.71828\dots \quad (16)$$

Finally, in all cases the transition from the matrix $W_{\mathcal{I}', \mathcal{J}'}^+$ to the matrix $U = W_{\mathcal{I}, \mathcal{J}}^+$ output by the algorithms of [GE96] increases the upper bound on the matrix norm by at most a factor of $\sqrt{((k-r)r+1)((l-r)r+1)}$ (cf. (13)), and so

$$\|U\| = \beta \|W_{\mathcal{I}', \mathcal{J}'}^+\|, \quad \text{for } \beta \leq \sqrt{((k-r)r+1)((l-r)r+1)}. \quad (17)$$

Remark 9. The above analysis provides some guidance for the choice of the integers k , $r \leq k \leq m$, and l , $r \leq l \leq n$. Namely we should keep the arithmetic cost of Algorithm 6, of order $(m+n)kl$, which is significantly smaller than that of the known CUR algorithms, but we should try to increase the ratio $\mathbb{E}(\sigma_r(W_{\mathcal{I}', \mathcal{J}'}))/\sigma_r(W)$ by increasing the integers k and l .

3.4 Error estimates for the CUR approximation of the average input

Now suppose that the matrix W is perturbed by a matrix E whose norm $\|E\|$ is sufficiently small, so that the above estimates for $\|U\|$ are little affected. Then Corollary 4, for $U' \approx U$, Remark 5, bound (7), and Theorem 49 combined enable us to estimate the spectral and Frobenius norms $|\Delta|$ of the error matrix of the output approximation of Algorithm 6.

Let us next specify such bounds in case where W is a scaled factor-Gaussian matrix with expected rank r ; for $\theta = 1$ this covers the special case where W is a factor-Gaussian matrix, $k = l = \alpha r$ for a constant $\alpha > 1$, and the value $\sigma_{r+1}(W)$ is small enough so that we can ignore the terms in $O(\epsilon^2)$ in (11) and (12) and the difference between U and U' . Furthermore estimating the Frobenius norm $\|\Delta\|_F$ and by scaling the matrix W , we can ensure that $\|U\|_F \gg \|W_{\mathcal{I}, \mathcal{J}}\|_F$. Then we obtain

- (i) $\|\Delta\|_F \leq 2\psi\sigma'_{r+1}(W)\|U\| \min\{\|C\|_F, \|R\|_F\}$ and
- (ii) $\|\Delta\| \leq \sigma_{r+1}(W)(2 + \kappa(W_{\mathcal{I}, \mathcal{J}}))\|U\| \min\{\|C\|, \|R\|\}$.

Here $\psi := \|U\|/\|U\|_F$, $1/r \leq \psi \leq 1$, $\kappa(W_{\mathcal{I}, \mathcal{J}}) = \|W_{\mathcal{I}, \mathcal{J}}\|/\|U\|$, $C = G_{m, r}H_{r, r}$, $R = G_{r, r}H_{r, n}$, $W_{\mathcal{I}, \mathcal{J}} = G_{r, r}H_{r, r}$, $F_{i, j} \in \mathcal{G}^{i \times j}$, $\|F_{i, j}\| = \nu_{i, j}$, $i, j \in \{m, n, r\}$, and F stands for G , H , or $W_{\mathcal{I}, \mathcal{J}}$.

Furthermore substitute the estimates of Theorem 47 and obtain

$$\mathbb{E}(\|C\|) \leq \mathbb{E}(\nu_{m, r})\mathbb{E}(\nu_{r, r}) \leq 2(\sqrt{mr} + r), \quad \mathbb{E}(\|R\|) \leq E(\nu_{r, r})E(\nu_{r, n}) \leq 2(\sqrt{nr} + r), \quad (18)$$

$$\mathbb{E}(\|W_{\mathcal{I}, \mathcal{J}}\|) \leq E(\nu_{r, r})^2 \leq 4r.$$

Apply Theorem 49 and obtain

$$\mathbb{E}(\|U\|) \leq \beta E(\nu_{k, r}^+)E(\nu_{r, l}^+) \leq \frac{e^2\alpha\beta}{(\alpha-1)^2r}, \quad (19)$$

for β of (17) and α of our choice.

In the case where W is an l-factor Gaussian matrix and we apply Algorithm 6 for $k = m$ according to the left/right orientation rule of the previous subsection, we extend the above estimates as follows:

$C = G_{m,r}\bar{H}_{r,r}$, $R = G_{r,r}H$, $W_{\mathcal{I},\mathcal{J}} = G_{r,r}\bar{H}_{r,r}$,
 $G_{i,j} \in \mathcal{G}^{i \times j}$, $\bar{H}_{r,r}$ is the submatrix of H output by algorithms of [GE96] applied to the matrix $W = H$, $\|\bar{H}_{r,r}\| \leq \|H\| \leq 1$, and so

$$\mathbb{E}(\|C\|) \leq \mathbb{E}(\nu_{m,r}) \leq \sqrt{m} + \sqrt{r}, \quad \mathbb{E}(\|R\|) \leq \mathbb{E}(\nu_{r,r}) \leq 2\sqrt{r}, \quad \mathbb{E}(\|W_{\mathcal{I},\mathcal{J}}\|) \leq \mathbb{E}(\nu_{r,r})^2 \leq 4r,$$

$$\|U\| = \|W_{\mathcal{I},\mathcal{J}}^+\| = \beta \|W_{\mathcal{I}',\mathcal{J}'}^+\| \leq \beta h_+ \mathbb{E}(\nu_{l,r}^+) \leq \frac{e\beta h_+ \sqrt{l}}{l-r}.$$

In the case where W is a r-factor Gaussian matrix, we apply Algorithm 6 for $l = n$ according to the left/right orientation rule and then deduce similar estimates.

By applying (2) we can obtain similar estimates in terms of the Frobenius norm.

3.5 Á póstèrióri estimation and computation of the output errors

We can estimate the spectral and Frobenius norms of the matrices $C = W_{\cdot,\mathcal{J}}$, $R = W_{\mathcal{I},\cdot}$, and $U = W_{\mathcal{I},\mathcal{J}}^+$ by performing $O((m+n)r + r^3)$ flops. If the matrix W has numerical rank r and the value $\sigma'_{r+1}(W)$ is small, we can substitute these estimates into bound (7), Corollary 4, and Remark 5, and obtain some upper bounds on the ratios $\frac{|\Delta(r)|}{\sigma'_{r+1}(W)}$ and $\frac{|\Delta|}{\sigma'_{r+1}(W)} \leq \frac{|\Delta(r)|}{\sigma'_{r+1}(W)} + 1$.

Alternatively we can directly estimate the output errors and as by-product verify the hypothesis that the value $\sigma'_{r+1}(W)$ is small.

Algorithm 10. *Computation of the output errors of Algorithm 6.*

INPUT: Five integers k , l , m , n , and r , such that $1 \leq r \leq k \leq m$ and $r \leq l \leq n$, a small positive tolerance value ϵ , an $m \times n$ matrix W having numerical rank r , and the subsets \mathcal{I} of its k row indices and \mathcal{J} of its l column indices.

OUTPUT: Either SUCCESS and an upper bound on norm of the output error matrix or FAILURE.

COMPUTATIONS:

1. Compute three matrices C , U and R defining a CUR approximation $\tilde{W} = CUR$ of the matrix W .
2. Compute the error matrix $\tilde{\Delta} = \tilde{W} - W$.
3. Compute the spectral or Frobenius norm $|\tilde{\Delta}|$.
Output SUCCESS if $\delta \geq |\tilde{\Delta}|$. Output FAILURE otherwise.

We use $O((m+n)kl)$ flops at stage 1 and mn subtractions at stage 2. At stage 3 we can either compute the square of the Frobenius norm $\|\tilde{\Delta}\|_F^2$ by using $2mn - 1$ flops or estimate from above the spectral norm $\|\tilde{\Delta}\|$. We can do this in two steps:

(i) first compute one of the norms $\|\tilde{\Delta}\|_1$ (by using $(m-1)n$ flops and $n-1$ comparisons), $\|\tilde{\Delta}\|_\infty$ (by using $(n-1)m$ flops and $m-1$ comparisons), both of them (by using $mn - m - n$ flops and $m + n - 2$ comparisons), or $\|\tilde{\Delta}\|_C$ (by using $mn - 1$ comparisons) and

(ii) then apply (1), (2), or Remark 48.

If $kl \ll m+n$, then the dominant part of the overall complexity of the algorithm is about $2mn$ or $3mn$ flops and comparisons at stages 3 and 4. These bounds greatly exceed the estimated computational cost of performing Algorithm 6, but are dominated by orders of $nm \min\{m, n\}$ or $mn \log(n)$ flops involved in the algorithms of [GE96] and [P00] or [BW14], respectively.

3.6 Fast randomized á póstèríóri estimation of the output errors

We can dramatically accelerate Algorithm 10 if we test whether $\|\tilde{\Delta}'\| \leq \delta'$ is satisfied just for a random submatrix $\tilde{\Delta}'$ of a fixed smaller size (e.g., for the matrix $\tilde{\Delta}' = W_{\mathcal{I}', \mathcal{J}'} - W_{\mathcal{I}', \mathcal{J}'} U W_{\mathcal{I}, \mathcal{J}'}$ where the sets \mathcal{I} and \mathcal{J} and the matrix U are output by Algorithm 6) and a fixed tolerance value $\delta' \leq \delta$ instead of testing the bound $\|\tilde{\Delta}\| \leq \delta'$.

If $\|\tilde{\Delta}'\| > \delta'$, we output FAILURE. Otherwise we output PASS($\tilde{\Delta}', \delta'$).

For the worst case input matrix W , PASS($\tilde{\Delta}', \delta'$) implies SUCCESS of Algorithm 6 only if $\tilde{\Delta}' = \tilde{\Delta}$ and $\delta = \delta'$, but then we come back to more costly Algorithm 10.

For some input classes, however (e.g., where the deviation of the matrix W from its rank r approximation is due to errors of measurement or rounding), we can assume that the entries of the error matrix Δ are the observed i.i.d. values of a single random variable. If the number of the entries is large enough, say, exceeds 100, then by virtue of the Central Limit Theorem we can safely assume that the distribution of the variable is Gaussian (cf., e.g., [AW07]).

Then for a pair of integers q and s such that the product qs is in $O((m+n)kl)$ and thus is much less than mn , but is large enough (exceeding 100), we can apply our tests just to a random $q \times s$ submatrix Δ' of the $m \times n$ error matrix Δ .

Under this choice we compute the error matrix at a dominated arithmetic cost in $O((m+n)kl)$, but we still yield verification with high confidence, by applying customary rules of *hypothesis testing for the variance of a Gaussian variable*.

Indeed suppose that we have observed the values g_1, \dots, g_K of a Gaussian random variable g with a mean value μ and variance σ^2 and that we have computed the observed average value $\bar{\mu}_K = \frac{1}{K} \sum_{i=1}^K |g_i|$ and the observed variance $\sigma_K^2 = \frac{1}{K} \sum_{i=1}^K |g_i - \bar{\mu}_K|^2$. In this case, for a fixed reasonably large K , both Probability $\{|\bar{\mu}_K - \mu| \geq t|\mu|\}$ and Probability $\{|\sigma_K^2 - \sigma^2| \geq t\sigma^2\}$ converge to 0 exponentially fast as t grows to the infinity (cf., e.g., [C46]).

3.7 A class of hard inputs for Algorithm 6

Our analysis of Algorithm 6 does not apply to the worst case input W , for which the algorithm has an obvious weakness: it never visits the entries outside the selected $k \times l$ submatrix $W_{\mathcal{I}', \mathcal{J}'} = W_{\mathcal{I}', :, :}$, and for various choices of these entries the algorithm produces a poor CUR approximation of W .

Example 11. Consider $m \times n$ matrices $W = \text{diag}(Q \mid O_{m, n-r}) + E$ where Q is an $m \times r$ unitary matrix and E is an $m \times n$ matrix of full rank such that $\|E\| \ll 1$. Suppose that we apply Algorithm 6 to this matrix and choose the set $\mathcal{J}' \subseteq \{r+1, r+2, \dots, n\}$. In this case the algorithm produces a matrix $W_{\mathcal{I}, \mathcal{J}'}$ and a basic matrix $U = W_{\mathcal{I}, \mathcal{J}'}^+$ independent of the submatrix Q of W . The resulting CUR approximation expresses all m rows of the submatrix Q (up to a perturbation matrix Δ_Q) as linear combinations of its r rows indexed by the set \mathcal{I} . These linear combinations are defined independently of the matrix Q , and so, clearly, the norm $\|\Delta_Q\|$ can be small only for a small subset of $m \times r$ unitary matrices Q .

How likely would a set \mathcal{J}' lie in $\{r+1, r+2, \dots, n\}$? There are $\binom{n-r}{l}$ such bad choices,

and they make up the fraction $\prod_{j=0}^{r-1} \frac{n-j}{n-l-j}$ among all $\binom{n}{l}$ possible choices. This fraction converges to 1 as the ratio l/n converges to 0.

A similar argument shows a similar property of matrices $W = \begin{pmatrix} V \\ O_{m-r, n} \end{pmatrix} + E$ where V is

a $r \times n$ unitary matrix and E is an $m \times n$ matrix of full rank such that $\|E\| \ll 1$. Algorithm 6 produces poor CUR approximation of such matrices W for the fraction $f = \prod_{j=0}^{r-1} \frac{m-j}{m-k-j}$ made up by bad choices of a set $\mathcal{I}' \subseteq \{r+1, r+2, \dots, n\}$ among all $\binom{m}{k}$ possible choices, and $f \rightarrow 1$ as $k/m \rightarrow 0$.

The algorithm would still perform poorly on most of the inputs if we mask the patterns of this example by pre- and post-multiplying the input matrix W by permutation matrices P and P' , respectively, and simultaneously map the initial sets of indices as follows: $\mathcal{I}' \rightarrow P\mathcal{I}'$ and $\mathcal{J}' \rightarrow P'\mathcal{J}'$.

4 CUR cross-approximation: Generic algorithm and a class of hard inputs

For a non-costly remedy against the difficulty revealed by Example 11 we can extend Algorithm 6: we can first re-apply it to the matrix $W_{:, \mathcal{J}_1}$ for its output index set $\mathcal{J}_1 = \mathcal{J}$ of columns, then re-apply it again to the matrix $W_{\mathcal{I}_1, :}$ for the index set $\mathcal{I} = \mathcal{I}_1$ of rows output in this new application of the algorithm, and so on, arriving at recursive *CUR cross-approximation algorithms* (cf. [GOSTZ10]). Next we describe them in generic form and later analyze and elaborate upon them.

Algorithm 12. *Generic CUR cross-approximation.*

- INPUT: (i) Two positive tolerance values – an integer t and a small real τ ,
(ii) a subclass $\mathcal{W}_{m,n}$ in the space of $m \times n$ matrices W such that $W = \tilde{W} + E$, $r := \text{rank}(\tilde{W}) \ll \min\{m, n\}$, and $\|E\| \leq \epsilon$, for an unknown positive ϵ and two unknown matrices E and \tilde{W} , and
(iii) a *Basic Subalgorithm* (e.g., Algorithm 6), with its fixed *Stopping Criterion*, producing a $r \times r$ submatrix of a $k \times n$ or $m \times l$ matrix, for $k \geq r, l \geq r$.

INITIALIZATION: Fix an integer k such that $r \leq k < m$ and an index set \mathcal{I}_0 of k rows of the matrix W . Set $s = 0$.

OUTPUT: FAILURE or a $r \times r$ submatrix $W_{\mathcal{I}, \mathcal{J}}$ defining a CUR approximation of the matrix W such that $\|\Delta\| \leq \tau$ for Δ of (7).

- COMPUTATIONS: 1. If $s > t$, then stop and output FAILURE. Apply the Basic Subalgorithm to the matrix $W_{\mathcal{I}_s, :}$; output a $k \times l$ submatrix $W_{\mathcal{I}_s, \mathcal{J}_{s+1}}$. If $s \geq 1$ and if setting $W_{\mathcal{I}, \mathcal{J}} = W_{\mathcal{I}_s, \mathcal{J}_{s+1}}$ defines CUR approximation of the matrix W within a fixed tolerance τ to the norm $\|\Delta\|$, then stop and output the matrix $W_{\mathcal{I}, \mathcal{J}} = W_{\mathcal{I}_s, \mathcal{J}_{s+1}}$. Otherwise go to stage 2.
2. Apply the Basic Subalgorithm to the matrix $W_{:, \mathcal{J}_{s+1}}$; output a $k \times l$ submatrix $W_{\mathcal{I}_{s+1}, \mathcal{J}_{s+1}}$. If setting $W_{\mathcal{I}, \mathcal{J}} = W_{\mathcal{I}_{s+1}, \mathcal{J}_{s+1}}$ defines CUR approximation of the matrix W within a fixed tolerance τ to the norm $\|\Delta\|$, then stop and output the matrix $W_{\mathcal{I}, \mathcal{J}} := W_{\mathcal{I}_{s+1}, \mathcal{J}_{s+1}}$. Otherwise go to stage 1 for the integer $s + 1$ replacing s .

Remark 13. We can immediately define a dual version of the algorithm, by initializing it with an integer l , for $r \leq l \leq n$, and an index set \mathcal{J}_0 of l columns of the matrix W , and then dealing with matrices $W_{:, \mathcal{J}_s}$ and $W_{\mathcal{I}_{s+1}, \mathcal{J}_s}$ rather than matrices $W_{\mathcal{I}_s, :}$ and $W_{\mathcal{I}_s, \mathcal{J}_{s+1}}$.

We can invoke stage 1 or 2 of Algorithm 12 or its dual version of Remark 13 by applying Algorithm 6 for $l = n$ or $k = m$, respectively, and can immediately extend our error analysis of the previous section. Moreover, unlike Algorithm 6, Algorithm 12 supports close CUR approximations of most of the matrices W of Example 11, although not the matrices of the following narrower class.

Example 14. Apply Algorithm 12 to an $m \times n$ matrix $W = \text{diag}(U, E)$ where U is a $r \times r$ unitary matrix and E is a $(m-r) \times (n-r)$ matrix of full rank such that $\|E\| \ll 1$. Suppose that $\mathcal{I}_0 \subseteq \{r+1, r+2, \dots, m\}$ and observe that in this case $\mathcal{I}_s \subseteq \{r+1, r+2, \dots, m\}$ and $\mathcal{J}_s \subseteq \{r+1, r+2, \dots, n\}$ for all s , in particular for the sets \mathcal{I} and \mathcal{J} defining the submatrix $W_{\mathcal{I}, \mathcal{J}}$ of W output by Algorithm 12. Therefore the resulting CUR approximation is poor: it is a block diagonal matrix of the form $\text{diag}(O_{r,r}, X)$, and so $\|W - CUR\| \geq 1$. Similarly to Example 11, estimate that $\mathcal{I}_0 \subseteq \{r+1, r+2, \dots, m\}$ with frequency $\prod_{j=0}^{r-1} \frac{m-j}{m-k-j}$, which converges to 1 as the ratio k/m converges to 0. Likewise, the dual algorithm of Remark 13 produces a poor CUR approximation if it begins with a set of l integers such that $\mathcal{J}_0 \in \{r+1, r+2, \dots, n\}$, and in the set of all l -tuples of integers in $\{1, \dots, n\}$ such bad sets make up the fraction $\prod_{j=0}^{r-1} \frac{n-j}{n-l-j}$, which converges to 1 as the ratio l/n converges to 0. Algorithm 12 would still perform poorly for most of inputs if we mask the patterns of this example as at the very end of Section 3.7.

5 Cross-approximation and volume maximization

5.1 Section outline

We can bound the output errors of Algorithm 12 according to our analysis in Section 2, but Theorems 16 and 17 of the next subsection support stronger bounds if we can compute a $k \times l$ submatrix $W_{\mathcal{I}, \mathcal{J}}$ of an $m \times n$ input matrix W whose volume or the r -projective volume (these concepts are defined by (20)) is h -maximal, that is, maximal within a reasonably bounded factor h among all $k \times l$ submatrices of W .

This motivates search for such a $k \times l$ submatrix. Our Theorems 26 and 27 imply if a matrix W is close to a rank- r matrix, then it is sufficient to compute a submatrix with h -maximal volume among $k \times l$ submatrices lying in two fixed sets of k rows and l columns of W , respectively.

Even for this simplified task only semi-heuristic algorithms are known, but we estimate that any $k \times l$ submatrix $W_{\mathcal{I}, \mathcal{J}}$ of the average $m \times l$ or $k \times n$ matrix has such a nearly maximal volume.

We next recall the basic concepts of the volume and the r -projective volume of a matrix, then prove the cited theorems, recall the relevant known algorithms, and comment on adjusting generic CUR algorithm accordingly.

5.2 The volume and the r -projective volume of a matrix and the impact of their maximization on the output approximation

For a triple of integers k, l , and r such that $1 \leq r \leq \min\{k, l\}$, the *volume* $v_2(M)$ and the *r -projective volume* $v_{2,r}(M)$ of a $k \times l$ matrix M are defined as follows:

$$v_2(M) := \prod_{j=1}^{\min\{k, l\}} \sigma_j(M), \quad v_{2,r}(M) := \prod_{j=1}^r \sigma_j(M), \quad (20)$$

$$v_{2,r}(M) = v_2(M) \text{ if } r = \min\{k, l\}, \quad (21)$$

$$v_2^2(M) = \det(MM^*) \text{ if } k \geq l; \quad v_2^2(M) = \det(M^*M) \text{ if } k \leq l, \quad v_2^2(M) = |\det(M)| \text{ if } k = l.$$

These concepts can be traced back to [B-I92] and are basic for computing CUR approximation in [B-I92], [CI94], [GZT95], [GTZ97], [GTZ97a], [T96], [T00], [GT11], [M14], and [O16]. In particular the following results from [O16] extend the study in [GZT95], [GTZ97], [GTZ97a], [GT01], [GT11], and [M14].

Theorem 15. *Suppose that $W_{\mathcal{I},\mathcal{J}}$ is a nonsingular $r \times r$ submatrix of an $m \times n$ matrix W and*

$$hv_2(W_{\mathcal{I},\mathcal{J}}) = \max_B v_2(B),$$

for a fixed $h \geq 1$, where the maximum is over all $r \times r$ submatrices. Then

$$\|\Delta\|_C \leq (r+1)h \sigma_{r+1}(W)$$

for $\Delta' := W - \bar{W}$ of (7) and \bar{W} of equations (3) and (4).

This theorem has been proven in [GT01, Corollary 2.3] in the case where $m = n$. It is the special case (where $k = l = r$ and $m = n$) of the following theorem.

Theorem 16. *[O16, Theorem 6]. Suppose that $W_{\mathcal{I},\mathcal{J}}$ is a $k \times l$ submatrix of an $m \times n$ matrix W and*

$$hv_2(W_{\mathcal{I},\mathcal{J}}) = \max_B v_2(B),$$

for a fixed $h \geq 1$, where the maximum is over all submatrices B of the matrix W .

Let $r = \min\{k, l\}$ and define Δ and \bar{W} by equations (3), (4), and (7) (as in Theorem 15). Then

$$\|\Delta\|_C \leq h \sqrt{\frac{(k+1)(l+1)}{|l-k|+1}} \sigma_{r+1}(W).$$

Theorem 17. *[O16, Theorem 7]. Suppose that $W_{\mathcal{I},\mathcal{J}}$ is a $k \times l$ submatrix of a matrix W and*

$$hv_{2,r}(W_{\mathcal{I},\mathcal{J}}) = \max_B v_{2,r}(B) \text{ for some integer } r \text{ such that } 1 \leq r \leq \min\{k, l\},$$

for a fixed $h \geq 1$, where the maximum is over all $k \times l$ submatrices B of the matrix W . Define Δ by equations (3), (4), and (7) (as in Theorems 15 and 16), but write $U = (W'_{\mathcal{I},\mathcal{J}})^+$ where the matrix $W'_{\mathcal{I},\mathcal{J}}$ is defined by setting to 0 all singular values $\sigma_j(W_{\mathcal{I},\mathcal{J}})$ for $j > r$ (cf. (6)). Then

$$\|\Delta\|_C \leq h \sqrt{\frac{(k+1)(l+1)}{(k-r+1)(l-r+1)}} \sigma_{r+1}(W).$$

5.3 Which size is the best to choose for a basic matrix?

We first observe how the size of the submatrix $W_{\mathcal{I},\mathcal{J}}$ affects the estimates of Theorems 16 and 17 for its volume and r -projective volume.

(i) The bound of Theorem 16 turns into

$$\|\Delta\|_C \leq (r+1) h \sigma_{r+1}(W)$$

for $k = l = r$ and into

$$\|\Delta(r)\|_C \leq \sqrt{(1 + 1/b)(r + 1)} h \sigma_{r+1}(W)$$

if $k = r = (b + 1)l - 1$ or $l = r = (b + 1)k - 1$ and if $b > 0$. Therefore we improve by a factor of $\sqrt{\frac{k+1}{1+1/b}}$ the CUR approximation defined by a submatrix $W_{\mathcal{I},\mathcal{J}}$ of the maximal volume if we shift from choosing $k = l = r$ to choosing $k = r = (b + 1)l - 1$ or $l = r = (b + 1)k - 1$ for $b > 0$.

(ii) The bound of Theorem 17 turns into

$$\|\Delta\|_C \leq (1 + 1/b)h \sigma_{r+1}(W)$$

for $k = l = (b + 1)r - 1$ and a positive b . So the choice of $k = l = (b + 1)r - 1$ optimizes cross-approximation where we rely on a submatrix $W_{\mathcal{I},\mathcal{J}}$ that has the maximal r -projective volume.

The theorems show that is defined by a submatrix $W_{\mathcal{I},\mathcal{J}}$ that has maximal (up to a bounded factor h) volume or (even better) the r -projective volume defines a close CUR approximation. Next we show that we can maximize the r -projective volume as soon as we maximize the volume.

First notice that pre-multiplication of a $k \times n$ matrix W by a square unitary matrix does not affect the pairwise comparison of the r -projective volumes of its $k \times l$ submatrices for $r \leq k \leq n$.

Now suppose that $W = \bar{W} + E$, $\|E\|$ is small, and compute rank-revealing QRP factorization $W = QRP$ for unitary matrix Q , permutation matrix P , and $R = \begin{pmatrix} R' \\ R'' \end{pmatrix}$ where $\|R''\| \leq \|E\|$ (cf. [GL13, Sections 5.4.3 and 5.4.4] and [GE96]).⁷

Therefore we are motivated to apply the following step for computing a basic submatrix U of a CUR approximation of a $k \times n$ matrix W :

(i) choose the size of a basic submatrix according to the above rules for minimizing the norm $\|\Delta\|_C$ (based on Theorem 17),

(ii) compute rank-revealing QRP factorization $W = QRP$,

(iii) compute a submatrix of the matrix $\bar{R} = \begin{pmatrix} R' \\ O \end{pmatrix}$ and let the r rows of the matrix R' define the index set \mathcal{I} ,

(iv) maximize the volume $v_2(\bar{R})$ [this nearly maximizes the r -projective volume $v_{2,r}(R)$ because $v_2(\bar{R}) = v_2(R') = v_{2,r}(R') = v_{2,r}(\bar{R}) \approx v_{2,r}(R)$] and let the l columns of the computed matrix define the index set of columns \mathcal{J} , and finally

(v) compute a basic matrix $U = W_{\mathcal{I},\mathcal{J}}^+$.

Remark 18. The algorithm slightly simplifies the guiding rules of Theorem 17 by skipping the stage of setting to 0 the singular values $\sigma_j(W_{\mathcal{I},\mathcal{J}})$ for $j > r$; our tests show that this deviation does not worsen the resulting CUR approximation.

The error estimate for the output of the algorithm is superior to those of Corollary 4 (provided that the factor h is reasonably small) even we consider minimization of the Chebyshev norm a disadvantage and shift to a larger spectral or Frobenius norm bound according to (2) or Remark 48.

⁷We can apply other rank-revealing factorizations instead (cf. [GL13, Section 5.4.6]).

5.4 The volume and the r -projective volume of a matrix product

Theorem 19. (See Examples 20 and 21 and Remark 22 below.)

Suppose that $W = GH$ for an $m \times q$ matrix G and a $q \times n$ matrix H . Then

- (i) $v_2(W) = v_2(G)v_2(H)$ if $q \leq \min\{m, n\}$,
- (ii) $v_{2,r}(W) \leq v_{2,r}(G)v_{2,r}(H)$ for $r \leq q$,
- (iii) $v_2(G)v_2(H) \geq v_2(W) = 0$ if $\min\{m, n\} > q$, and
- (iv) $v_2(W) \leq v_2(G)v_2(H)$ if $m = n \leq q$.

The following examples show some limitations on possible extensions of the theorem.

Example 20. If G and H are unitary matrices and if $GH = O$, then $v_2(G) = v_2(H) = v_{2,r}(G) = v_{2,r}(H) = 1$ and $v_2(GH) = v_{2,r}(GH) = 0$ for all $r \leq q$.

Example 21. If $G = (1 \mid 0)$ and $H = \text{diag}(1, 0)$, then $v_2(G) = v_2(GH) = 1$ and $v_2(H) = 0$.

Remark 22. [O16] proves claims (i) (in the case where $m = q$ or $n = q$), (iii), and (iv) (by using a different proof). We do not use claim (iii), but include it for the sake of completeness.

Proof. We first prove claim (i).

Let $G = S_G \Sigma_G T_G^*$ and $H = S_H \Sigma_H T_H^*$ be SVDs such that Σ_G , T_G^* , S_H , Σ_H , and $U = T_G^* S_H$ are $q \times q$ matrices and S_G , T_G^* , S_H , T_H^* , and U are unitary matrices.

Write $V = \Sigma_G U \Sigma_H$ and notice that $\det(V) = \det(\Sigma_G) \det(U) \det(\Sigma_H)$ where $|\det(U)| = 1$ because U is a square unitary matrix.

Hence $v_2(V) = |\det(V)| = |\det(\Sigma_G) \det(\Sigma_H)| = v_2(G)v_2(H)$.

Now let $V = S_V \Sigma_V T_V^*$ be SVD where S_V , Σ_V and T_V^* are $q \times q$ matrices and S_V and T_V^* are unitary matrices.

Observe that $W = S_G V T_H^* = S_G S_V \Sigma_V T_V^* T_H^* = S_W \Sigma_W T_W^*$ where $S_W = S_G S_V$ and $T_W^* = T_V^* T_H^*$ are unitary matrices. Consequently $W = S_W \Sigma_W T_W^*$ is SVD, and so $\Sigma_W = \Sigma_V$.

Therefore $v_2(W) = v_2(V) = v_2(G)v_2(H)$, which proves claim (i).

Next we prove claim (ii).

First assume that $q \leq \min\{m, n\}$ as in claim (i) and let $W = S_W \Sigma_W T_W^*$ be SVD.

In this case we have proven that $\Sigma_W = \Sigma_V$ for $V = \Sigma_G U \Sigma_H$, $q \times q$ diagonal matrices Σ_G and Σ_H , and a $q \times q$ unitary matrix U . Consequently $v_{2,r}(W) = v_{2,r}(\Sigma_V)$.

In order to prove claim (ii) in the case where $q \leq \min\{m, n\}$, it remains to deduce that

$$v_{2,r}(\Sigma_{r,V}) \leq v_{2,r}(G)v_{2,r}(H). \quad (22)$$

Notice that $\Sigma_V = S_V^* V T_V = S_V^* \Sigma_G U \Sigma_H T_V$ for $q \times q$ unitary matrices S_V^* and H_V .

Let $\Sigma_{r,V}$ denote the $r \times r$ leading submatrix of Σ_V , and so $\Sigma_{r,V} = \widehat{G} \widehat{H}$ where $\widehat{G} := S_{r,V}^* \Sigma_G U$ and $\widehat{H} := \Sigma_H T_{r,V}$ and where $S_{r,V}$ and $T_{r,V}$ denote the $r \times q$ leftmost unitary submatrices of the matrices S_V and T_V , respectively.

Observe that $\sigma_j(\widehat{G}) \leq \sigma_j(G)$ for all j because \widehat{G} is a submatrix of the $q \times q$ matrix $S_V^* \Sigma_G U$, and similarly $\sigma_j(\widehat{H}) \leq \sigma_j(H)$ for all j . Therefore $v_{2,r}(\widehat{G}) = v_2(\widehat{G}) \leq v_{2,r}(G)$ and $v_{2,r}(\widehat{H}) = v_2(\widehat{H}) \leq v_{2,r}(H)$. Also notice that $v_{2,r}(\Sigma_{r,V}) = v_2(\Sigma_{r,V})$.

Furthermore $v_2(\Sigma_{r,V}) = v_2(\widehat{G})v_2(\widehat{H})$ by virtue of claim (i) because $\Sigma_{r,V} = \widehat{G} \widehat{H}$.

Combine the latter relationships and obtain (22), which implies claim (ii) in the case where $q \leq \min\{m, n\}$.

Next we extend claim (ii) to the general case of any positive integer q .

Embed a matrix H into a $q \times q$ matrix $H' = (H \mid O)$ banded by zeros if $q > n$. Otherwise write $H' = H$. Likewise embed a matrix G into a $q \times q$ matrix $G' = (G^* \mid O)^*$ banded by zeros if $q > m$. Otherwise write $G' = G$.

Apply claim (ii) to the $m' \times q$ matrix G' and $q \times n'$ matrix H' where $q \leq \min\{m', n'\}$.

Obtain that $v_{2,r}(G'H') \leq v_{2,r}(G')v_{2,r}(H')$.

Substitute equations $v_{2,r}(G') = v_{2,r}(G)$, $v_{2,r}(H') = v_{2,r}(H)$, and $v_{2,r}(G'H') = v_{2,r}(GH)$, which hold because the embedding keeps invariant the singular values and therefore keeps invariant the projective volumes of the matrices G , H , and GH . This completes the proof of claim (ii), which implies claim (iv) because $v_2(V) = v_{2,n}(V)$ if V stands for G , H or GH and if $m = n \leq q$.

Claim (iii) is trivially verified. \square

5.5 Volume maximization in cross-approximation of a rank- r matrix

We begin with some definitions and simple auxiliary results.

Definition 23. The volume of a $k \times l$ submatrix $W_{\mathcal{I},\mathcal{J}}$ of a matrix W is *h-maximal* if over all its $k \times l$ submatrices this volume is maximal up to a factor of h (cf. [P00]). The volume $v_2(W_{\mathcal{I},\mathcal{J}})$ is *column-wise* (resp. *row-wise*) *h-maximal* if it is *h-maximal* in the submatrix $W_{\mathcal{I},:}$ (resp. $W_{:, \mathcal{J}}$). Such a volume is *locally column-wise* (resp. *row-wise*) *h-maximal* if it is *h-maximal* over all submatrices of W that differ from it by a single column (resp. single row). We call volume (h_c, h_r) -*maximal* if it is both column-wise h_c -maximal and row-wise h_r -maximal. Likewise we define *locally* (h_c, h_r) -*maximal* volume. We call 1-maximal and $(1, 1)$ -maximal volumes *maximal*. The definitions can be restated in terms of the r -projective volume.

By comparing SVDs of two matrices W and W^+ we obtain the following result.

Lemma 24. $\sigma_j(W)\sigma_j(W^+) = 1$ for all subscripts j and all matrices W .

Corollary 25. $v_2(W)v_2(W^+) = 1$ and $v_{2,r}(W)v_{2,r}(W^+) = 1$ for all matrices W and all integers r such that $1 \leq r \leq \text{rank}(W)$.

The following theorem implies that maximization of the volume or the r -projective volume of a submatrix can be reduced to two stages of successive row and column maximization.

Theorem 26. *If the volume (resp. r -projective volume) of a $k \times l$ submatrix $W_{\mathcal{I},\mathcal{J}}$ is (h_c, h_r) -maximal in a matrix W of rank k or l , then the volume (resp. r -projective volume) of this submatrix is $h_c h_r$ -maximal.*

Proof. Write $k := |\mathcal{I}|$ and $l := |\mathcal{J}|$. Fix any $k \times l$ submatrix $W_{\mathcal{I}',\mathcal{J}'}$ of the matrix W , express it by applying equation (3) for $\tilde{W} := W_{\mathcal{I}',\mathcal{J}'}$, and obtain that

$$W_{\mathcal{I}',\mathcal{J}'} = W_{\mathcal{I}',\mathcal{J}} W_{\mathcal{I},\mathcal{J}}^+ W_{\mathcal{I},\mathcal{J}'}.$$

Suppose that $k \geq l$ and apply Theorem 19 at first for $G = W_{\mathcal{I}',\mathcal{J}}$ and $H = W_{\mathcal{I},\mathcal{J}}^+$ and then for $G = W_{\mathcal{I}',\mathcal{J}} W_{\mathcal{I},\mathcal{J}}^+$ and $H = W_{\mathcal{I},\mathcal{J}'}$ and obtain that $v(W_{\mathcal{I}',\mathcal{J}} W_{\mathcal{I},\mathcal{J}}^+ W_{\mathcal{I},\mathcal{J}'}) \leq v(W_{\mathcal{I}',\mathcal{J}})v(W_{\mathcal{I},\mathcal{J}}^+)v(W_{\mathcal{I},\mathcal{J}'})$ where v can stand for v_2 as well as for $v_{2,r}$ for any $r \leq l$.

Similarly obtain the same bound if $k \leq l$.

Recall Corollary 25 and obtain $v(W_{\mathcal{I}',\mathcal{J}} W_{\mathcal{I},\mathcal{J}}^+ W_{\mathcal{I},\mathcal{J}'}) \leq v(W_{\mathcal{I}',\mathcal{J}})v(W_{\mathcal{I},\mathcal{J}'})/v(W_{\mathcal{I},\mathcal{J}})$.

Recall that the matrix $W_{\mathcal{I},\mathcal{J}}$ is (h_c, h_r) -maximal and obtain that $v(W_{\mathcal{I},\mathcal{J}}) \geq h_c v(W_{\mathcal{I},\mathcal{J}'})$ and $v(W_{\mathcal{I},\mathcal{J}}) \geq h_r v(W_{\mathcal{I}',\mathcal{J}})$.

Substitute these inequalities into the above bound on ρ and obtain that $v(W_{\mathcal{I}',\mathcal{J}'}) \leq h_c h_r v(W_{\mathcal{I},\mathcal{J}})$ for any $k \times l$ submatrix of the matrix W . \square

5.6 The volume and the r -projective volume of a perturbed matrix

Next we estimate the impact of a perturbation E on the volume and the r -projective volume of a rank- r matrix M where $\|E\|$ is small.

Theorem 27. The Volume and the r -Projective Volume of a Perturbed Matrix. *Suppose that \tilde{M} and E are $k \times l$ matrices, $M = \tilde{M} + E$, $\text{rank}(\tilde{M}) = r \leq \min\{k, l\}$, and $\|E\| \leq \epsilon \leq \sigma_{r+1}(M)$. Then*

$$\left(1 - \frac{\sigma_{r+1}(M)}{\sigma_r(M)}\right)^r \leq \prod_{j=1}^r \left(1 - \frac{\epsilon}{\sigma_j(M)}\right) \leq \frac{v_{2,r}(M)}{v_{2,r}(\tilde{M})} \leq \prod_{j=1}^r \left(1 + \frac{\sigma_{r+1}(M)}{\sigma_j(M)}\right) \leq \left(1 + \frac{\epsilon}{\sigma_r(M)}\right)^r. \quad (23)$$

If $\min\{k, l\} = r$, then $v_2(M) = v_{2,r}(M)$, $v_2(\tilde{M}) = v_{2,r}(\tilde{M})$, and

$$\left(1 - \frac{\sigma_{r+1}(M)}{\sigma_r(M)}\right)^r \leq \frac{v_2(M)}{v_2(\tilde{M})} = \frac{v_{2,r}(M)}{v_{2,r}(\tilde{M})} \leq \left(1 + \frac{\sigma_{r+1}(M)}{\sigma_r(M)}\right)^r. \quad (24)$$

Proof. Bounds (23) follow because a perturbation of a matrix within a norm bound ϵ changes its singular values by at most ϵ (cf. [GL13, Corollary 8.6.2]). Bounds (24) follow because $v_2(N) = \prod_{j=1}^r \sigma_j(N)$ for any $k \times l$ matrix N with $\min\{k, l\} = r$, in particular for $N = M$ and $N = \tilde{M} + E$. \square

5.7 Estimation of output errors of cross-approximation with volume maximization

Suppose that we apply the first loop of cross-approximation Algorithm 12 with h_c -maximization of the volume at stage 1 and its h_r -maximization at stage 2. Then we compute a $k \times l$ submatrix $W_{\mathcal{I}_1, \mathcal{J}_1}$ of an input matrix W at the overall cost of performing $O((m+n)kl)$ flops. (We arrive at similar estimates if we maximize the r -projective volumes of $k \times l$ submatrices for $r \leq \min\{k, l\}$ rather than their volumes for $r = \min\{k, l\}$.)

If $\text{rank}(W) = r = \min\{k, l\}$, then by virtue of Theorem 26, the submatrix $W_{\mathcal{I}_1, \mathcal{J}_1}$ has the $h_c h_r$ -maximal volume in W , and we write $W_{\mathcal{I}, \mathcal{J}} := W_{\mathcal{I}_1, \mathcal{J}_1}$ and define a CUR decomposition of W .

If $\text{nrnk}(W) = r < \text{rank}(W)$, then such a matrix $W_{\mathcal{I}, \mathcal{J}}$ defines a CUR approximation of a matrix W . Section 2.2 enables us to bound its error norm, but Theorems 16, 17, 26, and 27 together enable sharper estimates if the ratio $\sigma_{r+1}(W)/\sigma_r(W_{\mathcal{I}, \mathcal{J}})$ is small. Namely we can obtain *a posteriori* estimates by readily computing this ratio for a $k \times l$ matrix $W_{\mathcal{I}, \mathcal{J}}$ if $r = \min\{k, l\}$.

Our next subject is *a priori* estimates.

We are going to show that the CUR approximation of Algorithm 12

(a) is close to the average input matrix W already when it is defined after a single loop of Algorithm 12, but

(b) is not close to the worst case input matrix W at any loop of Algorithm 12.

(a) Assume that an input matrix W is a perturbation $G\Sigma H + E$ of a scaled factor-Gaussian matrix $G\Sigma H$ with expected rank r where $G \in \mathcal{G}^{m \times r}$, $\Sigma \in \mathbb{C}^{r \times r}$, $H \in \mathcal{G}^{r \times n}$, and the norm $\|E\|$ is small enough, so that the perturbation by the matrix E little affects the volumes of well-conditioned submatrices of W . At this point recall that by virtue of Lemma 24 and Theorems 47 and 49 any submatrix of the matrix $G\Sigma H$ is likely to be well-conditioned (cf. (18) and Section 3.3). Now let us compare the volumes $v_2(G_{\mathcal{I}_0, \mathcal{J}_0} \Sigma H_{\mathcal{J}_0, \mathcal{J}'})$ where two sets \mathcal{I}_0 and \mathcal{J}_0 of cardinality r are fixed and the set \mathcal{J}' of cardinality r varies. In this case, by virtue of Theorem 19, $v_2(G_{\mathcal{I}_0, \mathcal{J}_0} \Sigma H_{\mathcal{J}_0, \mathcal{J}'}) = v_{2,G} v_2(\Sigma) v_{2,H}$ for $v_{2,G} = v_2(G_{\mathcal{J}_0, \mathcal{J}'})$ and $v_{2,H} = v_2(H_{\mathcal{J}_0, \mathcal{J}'})$. The factor $v_{2,G} \Sigma$ does not depend on the set \mathcal{J}_0 , and it remains to compare the volumes $v_{2,H} = v_2(H_{\mathcal{J}_0, \mathcal{J}'})$ for various sets \mathcal{J}' of cardinality r .

Recall that the real random variable $v_{2,H}$ is the product $\chi_r^2 \chi_{r-1}^2 \cdots \chi_1^2 = \prod_{j=1}^r \chi_j^2$ because $G_{\mathcal{J}_0, \mathcal{J}'} \in \mathcal{G}^{r \times r}$.

Furthermore let Γ denoting the Gamma function, recall that $\mathbb{E}(\chi_s^2) = s$ (in the real case) and that the probability density function (pdf) for χ_s^2 and $x \geq 0$ is given by

$$\frac{x^{0.5s-1} \exp(-0.5x)}{2^{0.5s} \Gamma(0.5s)},$$

and conclude that deviation of the volume $v_2 = v_2(G_{\mathcal{I}_0, \mathcal{J}_0} \Sigma H_{\mathcal{J}_0, \mathcal{J}'})$ from its expected value by a factor h greater than ζ occurs with a probability converging to 0 exponentially fast as ζ increases towards infinity. It follows that for the average input matrix W , already a random choice of the set \mathcal{J}' yields the maximal volume v_2 up to a strongly bounded factor h .

The above analysis is immediately extended to the special case of a small-norm perturbation $W = GH + E$ of a factor-Gaussian input GH with expected rank r , but it is also readily extended to the case of small-norm perturbations $W = GH + E$ of l- and r-factor-Gaussian inputs GH with expected rank r provided that the norm $\|E\|$ is small and that we apply the following *adjusted left/right orientation rule* (cf. Section 3.3):

- (i) apply Algorithm 12 if GH is an l-factor-Gaussian matrix with expected rank r , and
- (ii) apply its dual version of Remark 13 if GH is a r-factor-Gaussian matrix with expected rank r .

(b) No such favorable estimates apply to the worst case input, for which our cross-approximation algorithm can output poor CUR approximation. This occurs for a small fraction of inputs according to the above analysis in the case of a perturbed Gaussian input, but certainly occurs for the input matrices W of Example 14.

Indeed in the case of such an input our algorithm would turn into a long inefficient computational process, but we can detect such a development by monitoring the values of the volumes, the r -projective volumes, or the smallest positive singular values of the computed submatrices $W_{\mathcal{I}_s, \mathcal{J}_{s+1}}$ and $W_{\mathcal{I}_{s+1}, \mathcal{J}_{s+1}}$ for $s = 0, 1, \dots, s_0$. If we observe that these values stop changing or change too little as s increases, then we should stop the computations.

In view of Theorem 26 we should detect such a slowdown of the progress already for a small integer s , say, for $s \leq 4$ or even $s = 1$.

6 Contraction of the Class of Bad Inputs by Means of Preprocessing; Gaussian, Unitary, and Bidiagonal Factors

6.1 Gaussian preprocessing

Theorem 28. *Suppose that $G \in \mathcal{G}^{m \times m}$, $H \in \mathcal{G}^{n \times n}$, and V is an $m \times n$ well-conditioned matrix of rank r . Then, up to scaling by a constant,*

- (i) GV is an l -factor-Gaussian matrix of expected rank r ,
- (ii) VH is a r -factor-Gaussian matrix of expected rank r , and
- (iii) GVH is a scaled-factor Gaussian matrix of expected rank r .

Proof. Let $V = S_V \Sigma_V T_V^*$ be SVD where Σ_V is the diagonal matrix of the singular values of V , which is well-conditioned since so is V . Then

- (i) $GV = \bar{G} \Sigma_V T_V^*$,
- (ii) $VH = S_V \Sigma_V \bar{H}$, and
- (iii) $GVH = \bar{G} \Sigma_V \bar{H}$

where $\bar{G} = GS_V$ and $\bar{H} = T_V^* H$.

Both \bar{G} and \bar{H} are Gaussian matrices by virtue of rotational invariance of Gaussian matrices G and H because S_V and T_V are square unitary matrices and G and H are Gaussian matrices. Furthermore observe that $\kappa(\Sigma_V T_V^*) = \kappa(V) = \kappa(S_V \Sigma_V) = \kappa(\Sigma_V)$, and so the matrices $\Sigma_V T_V^*$, $S_V \Sigma_V$, and Σ_V are well-conditioned because so is the matrix V by assumption. \square

The theorem implies that our error analysis of application of Algorithms 6 and 12 to perturbations of l - and r -factor-Gaussian matrices of expected rank r and of scaled factor-Gaussian matrices can be applied to perturbations of the matrices GVH , VH , and GVH .

6.2 Simplified Preprocessing

Multiplication by square Gaussian matrices are expensive but next we show that by means of non-costly multiplication by sparse and structured matrices we can still narrow the class of bad inputs for our algorithms.

Indeed pre-multiply an l -factor-Gaussian $m \times n$ matrix of a small expected rank r (the study of r -factor-Gaussian and scaled factor-Gaussian matrices is similar) by $\bar{W} = GH$ for $G \in \mathcal{G}^{m \times r}$ by a unitary $m \times m$ matrix Q for $m \leq m$, and conclude by virtue of orthogonal invariance of Gaussian matrices that the $m \times n$ matrix QG is Gaussian and the matrix $Q\bar{W}$ is an l -factor Gaussian of expected rank r .

Suppose that $W = \bar{W} + E$ and the norm $\|E\|$ is small, apply Algorithms 6 and 12 to the small-norm perturbation $QW = Q\bar{W} + QE$ of the l -factor-Gaussian matrix $Q\bar{W}$, and extend our error analysis from the previous sections.

It follows that *Algorithms 6 and 12 produce close CUR approximations of the matrix $Q\bar{W} + QE$, $\bar{W} = GH$, defined as the average over all $m \times r$ Gaussian matrices G for every unitary multiplier Q .*

Let us estimate the arithmetic cost of this preprocessing. Algorithm 12 computes CUR approximation of the matrix QW , which involves only its k rows and l columns. Their computation involves $(2m - 1)kn + (2n - 1)ml$ flops if Q and W are dense unstructured matrices, but this cost bound decreases to less than $2(kn + ln)d$ if the multiplier Q has only d nonzero entries per row and per column. The families of such multipliers for small positive integers d , called Abridged Scaled Permuted Fourier and Hadamard (ASPF and ASPH)

matrices, have been applied in [PZ16] for preprocessing low-rank approximation algorithms, and extensive tests in [PZ16] have consistently shown efficiency of these multipliers.

Now assume that the matrix Q can be represented by using small memory space and can be multiplied by a vector fast and observe that these properties are extended from a CUR approximation of the matrix QW to the low-rank approximation Q^*CUR of the matrix W . In particular for any vector \mathbf{v} , we can reduce the computation of the vector $\mathbf{y} = Q^*CUR\mathbf{v}$ to the computation of the two vectors $\mathbf{x} = CUR\mathbf{v}$ and $\mathbf{y} = Q^*\mathbf{x}$.

For every multiplier Q there is a narrow class of $m \times n$ input matrices of numerical rank r for which one or both of Algorithms 6 and 12 fail to produce accurate CUR approximations, but for any multiplier Q such a class is narrow, and by intersecting the bad classes for a number of distinct multipliers Q we arrive at even a narrower class.

The impact of various policies of choosing multipliers on compression of this class is similar to that studied in [PZ16], and we refer the reader to that study. The same paper describes various classes of efficient well-conditioned and mostly unitary multipliers, to which we also refer the reader. The special cases of ASPF and ASPH multipliers seem to be most promising in our current applications, but we would also consider bidiagonal multipliers B .

Generally they are not unitary, but we choose them nonsingular and well-conditioned, and then it is not hard to extend our study of unitary multipliers to them, in particular to solve a linear system $\mathbf{y} = B\mathbf{v}$ for any vector \mathbf{v} .

6.3 Sparsification and incomplete factorization of a Gaussian matrix

Multiplication and inversion of Gaussian matrices are quite expensive, and for tentative simplification of these operations consider (i) sparsification of a Gaussian multiplier G and (ii) its incomplete factorization.

Case (i). Create a random sparse multiplier by setting to 0 sufficiently many entries of a Gaussian matrix G , so that any set of k rows and any set of l columns of the matrix $G_{\text{sparsified}}W$ and a vector $G_{\text{sparsified}}^{-1}\mathbf{v}$ for any fixed vector \mathbf{v} can be computed at a low cost, but keep sufficiently many entries of G untouched in order to have accuracy of CUR approximations of the matrix GW preserved in the transition $GW \rightarrow G_{\text{sparsified}}W$.

Case (ii). First decompose a Gaussian matrix G into a product of factors that can be readily inverted, then delete many of them so that we could compute at a low cost any set of k rows and any set of l columns of the product $F_1 \cdots F_h W$ of the remaining factors and an input matrix W .

One can examine a great variety of possible combinations of the known matrix factorizations (cf. [GL13], [S98]); here is a simple example.

Example 29. First compute PLU factorization of an $n \times n$ Gaussian matrix G by applying Gaussian elimination with row pivoting. Notice that $P^*G = LU$ is still a Gaussian matrix. Then readily decompose each of the triangular matrices L and U into the product of $n - 1$ bidiagonal matrices. Finally drop sufficiently many but not too many of the bidiagonal factors.

The above recipes seem to be quite promising in order to motivate their empirical study. Their formal support may be hard, but its attempts can provide useful insights and can be of independent theoretical interest. Next we present our first but nontrivial result in this direction: we prove that the probability distribution of the product of random and randomly

permuted cyclic bidiagonal matrices converges to Gaussian as the number of factors grows to the infinity.

Empirically, convergence to Gaussian distribution is quite fast (see Section 7.4), but we still cannot use these partial products as multipliers: a matrix that approximates a Gaussian matrix must be dense, and using it as a multiplier would imply expensive computations.

6.4 Approximation of a Gaussian matrix by a product of random bidiagonal matrices

Suppose that n is a positive integer, P is a random permutation matrix, and define $n \times n$ matrix

$$B := \begin{bmatrix} 1 & & & & \pm 1 \\ \pm 1 & 1 & & & \\ & \pm 1 & 1 & & \\ & & \dots & \dots & \\ & & & \pm 1 & 1 \end{bmatrix} P \quad (25)$$

where each ± 1 represents an independent Bernoulli random variable. Our goal is to prove the following theorem.

Theorem 30. *Let B_0, \dots, B_T be independent random matrices of the form (25). As $T \rightarrow \infty$, the distributions of the matrices $G_T := \prod_{t=1}^T B_t$ converge to the distribution of a Gaussian matrix.*

Before giving a proof consider a simpler scenario. Let

$$A := \begin{bmatrix} 1/2 & & & & 1/2 \\ 1/2 & 1/2 & & & \\ & 1/2 & 1/2 & & \\ & & \dots & \dots & \\ & & & & 1/2 \end{bmatrix} P \quad (26)$$

where P is a random permutation matrix. We are going to prove the following theorem.

Theorem 31. *Let A_0, \dots, A_T be independent random matrices defined by (26). As $T \rightarrow \infty$, the distributions of the matrices $\Pi_T := \prod_{t=1}^T A_t$ converge to the distribution of the matrix*

$$\begin{bmatrix} 1/n & \dots & \dots & 1/n \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 1/n & \dots & \dots & 1/n \end{bmatrix}.$$

Proof. First examine the effect of multiplying by such a random matrix A_i : let

$$M := (\mathbf{m}_1 \mid \mathbf{m}_2 \mid \dots \mid \mathbf{m}_n)$$

and let the permutation matrix of A_i defines a column permutation $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$. Then

$$MA_i = \left(\frac{\mathbf{m}_{\sigma(1)} + \mathbf{m}_{\sigma(2)}}{2} \mid \frac{\mathbf{m}_{\sigma(2)} + \mathbf{m}_{\sigma(3)}}{2} \mid \dots \mid \frac{\mathbf{m}_{\sigma(n)} + \mathbf{m}_{\sigma(1)}}{2} \right).$$

Here each new column is written as a linear combination of $\{\mathbf{m}_1, \dots, \mathbf{m}_n\}$. More generally, if we consider $MA_0A_1 \cdots A_t$, i.e., M multiplied with t matrices of the form (26), each new column has the following linear expression:

$$MA_0A_1 \cdots A_t = \left(\sum_k \pi_{k1}^t \mathbf{m}_k \mid \sum_k \pi_{k2}^t \mathbf{m}_k \mid \cdots \mid \sum_k \pi_{kn}^t \mathbf{m}_k \right).$$

Here π_{ij}^t is the coefficient of \mathbf{m}_i in the linear expression of the j th column of the product matrix.

Represent the column permutation defined by matrix of A_t by the map

$$\sigma_t : \{1, \dots, n\} \rightarrow \{1, \dots, n\};$$

then the following lemma clearly holds.

Lemma 32. *It holds that*

$$\sum_j \pi_{ij}^t = 1 \text{ for all } i$$

and

$$\pi_{ij}^{t+1} = \frac{1}{2}(\pi_{i\sigma_t(j)}^t + \pi_{i\sigma_t(j+1)}^t) \text{ for all pairs of } i \text{ and } j.$$

Now let us prove the following result.

Lemma 33. *For any i, j and $\epsilon > 0$,*

$$\lim_{T \rightarrow \infty} \text{Probability}(|\pi_{ij}^T - \frac{1}{n}| > \epsilon) = 0.$$

Proof. Fix i , define

$$\mathcal{F}^t := \sum_j (\pi_{ij}^t - \frac{1}{n})^2.$$

Then

$$\begin{aligned} \mathcal{F}^{t+1} - \mathcal{F}^t &= \sum_j (\pi_{ij}^{t+1} - \frac{1}{n})^2 - \sum_j (\pi_{ij}^t - \frac{1}{n})^2 \\ &= \sum_j \left[\left(\frac{\pi_{i\sigma_t(j)}^t + \pi_{i\sigma_t(j+1)}^t}{2} - \frac{1}{n} \right)^2 - \frac{1}{2} \left(\pi_{i\sigma_t(j)}^t - \frac{1}{n} \right)^2 - \frac{1}{2} \left(\pi_{i\sigma_t(j+1)}^t - \frac{1}{n} \right)^2 \right] \\ &= \sum_j \left[\left(\frac{\pi_{i\sigma_t(j)}^t + \pi_{i\sigma_t(j+1)}^t}{2} \right)^2 - \frac{1}{2} (\pi_{i\sigma_t(j)}^t)^2 - \frac{1}{2} (\pi_{i\sigma_t(j+1)}^t)^2 \right] \\ &\leq \sum_j -\frac{1}{4} \left[(\pi_{i\sigma_t(j)}^t)^2 - 2\pi_{i\sigma_t(j)}^t \pi_{i\sigma_t(j+1)}^t + (\pi_{i\sigma_t(j+1)}^t)^2 \right] \\ &= -\frac{1}{4} \sum_j (\pi_{i\sigma_t(j)}^t - \pi_{i\sigma_t(j+1)}^t)^2 \\ &\leq -\frac{1}{4n} \left(\sum_i |\pi_{i\sigma_t(j)}^t - \pi_{i\sigma_t(j+1)}^t| \right)^2 \\ &= -\frac{1}{4n} (\pi_{max}^t - \pi_{min}^t)^2. \end{aligned}$$

Here $\pi_{max}^t := \max_i \{\pi_{ij}^t\}$ and $\pi_{min}^t := \min_i \{\pi_{ij}^t\}$.

Furthermore, since $\pi_{max}^t \geq \pi_{ij}^t, \forall j$ and $\frac{1}{n} \geq \pi_{min}^t \geq 0$, it follows that

$$\mathcal{F}^t = \sum_j (\pi_{ij}^t - \frac{1}{n})^2 \leq n(\pi_{max}^t - \pi_{min}^t)^2.$$

Therefore

$$\mathcal{F}^{t+1} - \mathcal{F}^t \leq -\frac{1}{4n} (\pi_{max}^t - \pi_{min}^t)^2 \leq -\frac{1}{4n^2} \mathcal{F}^t.$$

Now our monotone decreasing sequence has the only stationary value when all values π_{ij}^t coincide with each other. Together with Lemma 33 this implies

$$\lim_{T \rightarrow \infty} \text{Probability}(|\pi_{ij}^T - \frac{1}{n}| > \epsilon) = 0.$$

□

Next we prove Theorem 32.

Proof. Let S_i^t denote the values ± 1 on each row. By definition $S_{i'}^t$ and S_i^t are independent for $i \neq i'$. Moreover, the following lemma can be readily verified.

Lemma 34. $\prod_{t=1}^T S_{i_t}^t$ and $\prod_{t=1}^T S_{i'_t}^t$ are independent as long as there is at least one index t such that $i_t \neq i'_t$.

Write

$$MB_0B_1 \cdots B_t = \left(\sum_k \gamma_{k1}^t \mathbf{m}_k \mid \sum_k \gamma_{k2}^t \mathbf{m}_k \mid \cdots \mid \sum_k \gamma_{kn}^t \mathbf{m}_k \right).$$

and notice that each γ_{ij}^T can be written as a sum of random values ± 1 whose signs are determined by $\prod_{t=1}^T S_{i_t}^t$. Since different signs are independent, we can represent γ_{ij}^T as the difference of two positive integers $\alpha - \beta$ whose sum is $2^T \pi_{ij}^T$.

Theorem 31 implies that the sequence π_{ij}^T converges to $\frac{1}{n}$ almost surely as $T \rightarrow \infty$.

Therefore $\gamma_{ij}^T / 2^T$ converges to Gaussian distribution as $T \rightarrow \infty$. Together with independence of the random values γ_{ij}^T for all pairs i and j , this implies that eventually the entire matrix converges to a Gaussian matrix (with i.i.d. entries). □

The speed of the convergence to Gaussian distribution is determined by the speed of the convergence (i) of $\lim_{T \rightarrow \infty} \pi_{ij}^T = 1$ and (ii) of the binomial distribution with the mean π_{ij}^t to the Gaussian distribution. For (i), we have the following estimate:

$$|\pi_{ij}^t - 1| \leq \mathcal{F}^t \leq (1 - \frac{1}{4n^2})^{t-1} \mathcal{F}^0;$$

and for (ii) we have the following Berry–Esseen Theorem:

Theorem 35. (Cf. [B41].) Let X_1, X_2, \dots be independent random variables with $E(X_i) = 0$, $E(X_i^2) = \sigma_i^2 > 0$ and $E(|X_i|^3) = \rho_i < \infty$ for all i . Furthermore let

$$S_n := \frac{X_1 + \cdots + X_n}{\sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2}}$$

be a normalized n -th partial sum. Let F_n and Φ denote the cumulative distribution functions of S_n a Gaussian variable, respectively. Then there exists a constant c such that for all n ,

$$\sup_{x \in \mathbb{R}} |F_n(x) - \Phi(x)| \leq c\psi,$$

where

$$\psi := \left(\sum_{i=1}^n \sigma_i^2 \right)^{-3/2} \max_{1 \leq i \leq n} \rho_i.$$

In our case, for any fixed i and j , write $N_t = 2^t \pi_{ij}^t$ and $\gamma_{ij}^t = X_1 + \dots + X_{N_t}$, where X_i are i.i.d. ± 1 variables. Then $E(X_i) = 0$, $E(X_i^2) = 1/4$, $E(X_i^3) = 1/8$, and

$$\begin{aligned} S_{N_t} &= \frac{X_1 + \dots + X_{N_t}}{\sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_{N_t}^2}} \\ &= \frac{\gamma_{ij}^t}{\sqrt{N_t/4}} \\ &= \frac{\sqrt{\pi_{ij}^t} \gamma_{ij}^t}{2^{t/2-1}}. \end{aligned}$$

Furthermore

$$\begin{aligned} \sup_{x \in \mathbb{R}} |F_N(x) - \Phi(x)| &\leq c \cdot \left(\sum_{i=1}^{N_t} \sigma_i^2 \right)^{-3/2} \max_{1 \leq i \leq N_t} \rho_i \\ &\leq \frac{1}{8} c \cdot \left(\frac{N_t}{4} \right)^{-3/2} \rightarrow 0 \text{ as } t \rightarrow \infty. \end{aligned}$$

7 Numerical Experiments

We tested the CGR algorithm on random input matrices as well as matrices of discretized partial differential equations (PDEs). We have run the tests in the Graduate Center of the City University of New York on a Dell computer with the Intel Core 2 2.50 GHz processor and 4G memory running Windows 7 and used MATLAB. In particular we applied its standard normal distribution function "randn()" in order to generate Gaussian matrices. We assumed that the numerical rank of the input matrices was given. Actually we calculated it by using the MATLAB's function "rank(-,1e-6)", which only counts singular values greater than 10^{-6} , and this means that approximation within error spectral norm bounds of 10^{-4} and even 10^{-3} can be considered reasonable for our matrix sizes.

7.1 CUR approximations of random input matrices

In this section we computed CUR approximations with random row- and column-selection for factored-Gaussian rank- r matrices, i.e., matrices A obtained from

$$A = A_1 * A_2 + \epsilon A_3,$$

for an $n \times r$ Gaussian matrix A_1 , an $r \times n$ Gaussian matrix A_2 , an $n \times n$ Gaussian matrix A_3 , and a small positive number ϵ . In the tests of this subsection we set $\epsilon = 10^{-10}$. (Recall that we call Gaussian a random matrix whose entries are i.i.d. standard Gaussian variables.)

In this subsection we compare the following four approaches:

- **Method 1:** Randomly choose two index sets \mathcal{I} and \mathcal{J} , then form a CUR approximation by

$$A_{approx} = A_{:, \mathcal{J}} \cdot A_{\mathcal{I}, \mathcal{J}}^{-1} \cdot A_{\mathcal{I}, :}.$$

- **Method 2:** Randomly choose a row index set \mathcal{I} , then perform a single loop of cross-approximation: pick column set \mathcal{J} of $A_{\mathcal{I}, :}$ such that $A_{\mathcal{I}, \mathcal{J}}$ has local maximum volume and pick a column set \mathcal{I} of $A_{:, \mathcal{J}}$ such that $A_{\mathcal{I}, \mathcal{J}}$ has local maximum volume. We did this by applying Alg.1 in [P00] and then defined a CUR approximation as in Method 1.

We also performed CUR decomposition of an $r \times r$ matrix with a $k \times l$ submatrix, where $k = l = 4r$.

- **Method 3:** Randomly choose a row index set \mathcal{K} and a column index set \mathcal{L} , both of cardinality $4r$, and then apply Algs. 1 and 2 from [P00] to compute a $r \times r$ submatrix $A_{\mathcal{I}, \mathcal{J}}$ within $A_{\mathcal{K}, \mathcal{L}}$ having local maximum volume. Define a CUR decomposition

$$A_{approx} = A_{:, \mathcal{J}} \cdot A_{\mathcal{I}, \mathcal{J}}^{-1} \cdot A_{\mathcal{I}, :}$$

- **Method 4:** Randomly choose a column index set \mathcal{L} ; then perform one round of row-update and column-update to increase the volume of $A_{\mathcal{K}, \mathcal{L}}$: First apply Alg. 1 from [P00] to find an index set \mathcal{K}' of size k such that $A_{\mathcal{K}', \mathcal{L}}$ has local maximal volume within $A_{:, \mathcal{L}}$, then apply Alg. 1 again (to matrix $A_{\mathcal{K}', :}$) to find an index set \mathcal{L}' of size l such that $A_{\mathcal{K}', \mathcal{L}'}$ has local maximal volume within $A_{\mathcal{K}', :}$. Then find an $r \times r$ submatrix $A_{\mathcal{I}, \mathcal{J}}$ having local maximal volume within $A_{\mathcal{K}', \mathcal{L}'}$.

Table 1 shows test results for all four methods for $n = 64, 128, 256, 512, 1024$ and $r = 8, 16, 32$; we performed 1000 runs for every setting. The table displays the mean and standard deviation of the error norm $\|A - A_{approx}\|_2$.

		Method 1		Method 2		Method 3		Method 4	
n	r	mean	std	mean	std	mean	std	mean	std
256	8	1.51e-05	1.40e-04	5.39e-07	5.31e-06	8.15e-06	6.11e-05	8.58e-06	1.12e-04
256	16	5.22e-05	8.49e-04	5.06e-07	1.38e-06	1.52e-05	8.86e-05	1.38e-05	7.71e-05
256	32	2.86e-05	3.03e-04	1.29e-06	1.30e-05	4.39e-05	3.22e-04	1.22e-04	9.30e-04
512	8	1.47e-05	1.36e-04	3.64e-06	8.56e-05	2.04e-05	2.77e-04	1.54e-05	7.43e-05
512	16	3.44e-05	3.96e-04	8.51e-06	1.92e-04	2.46e-05	1.29e-04	1.92e-05	7.14e-05
512	32	8.83e-05	1.41e-03	2.27e-06	1.55e-05	9.06e-05	1.06e-03	2.14e-05	3.98e-05
1024	8	3.11e-05	2.00e-04	4.21e-06	5.79e-05	3.64e-05	2.06e-04	1.49e-04	1.34e-03
1024	16	1.60e-04	3.87e-03	4.57e-06	3.55e-05	1.72e-04	3.54e-03	4.34e-05	1.11e-04
1024	32	1.72e-04	1.89e-03	3.20e-06	1.09e-05	1.78e-04	1.68e-03	1.43e-04	6.51e-04

Table 1: CUR approximation of random input matrices

7.2 CUR approximation of benchmark input matrices

In this section we tested algorithms with inputs from San Jose University Singular Matrix Database (<http://www.math.sjsu.edu/singular/matrices>). We have tested dense matrices with smaller ratios of "numerical rank/min(m, n)" for an $m \times n$ matrix. We have chosen matrices from the built-in test problems in Regularization Tools (see <http://www2.imm.dtu.dk/~pch/Regutools/> and the Regularization Tools Manual for references).

We tested three choices of the number r for row- and column-sampling: $r = \text{numerical rank} + 0, 4, 8$. In Table 2 we display the test results when we ran CUR algorithms (Methods 1 and 2) 10 times for every matrix. For Method 2 we used 5 cross-multiplications for each run, and for our real world inputs Method 2 has produced significantly more reliable CUR approximations, thus demonstrating the power of cross-approximation.

Inputs				Method 1		Method 2	
	m	n	r	mean	std	mean	std
baart	1000	1000	4	3.64e-02	6.53e-02	5.25e-04	1.49e-06
	1000	1000	8	2.07e-03	3.36e-03	7.43e-09	2.30e-09
	1000	1000	12	6.51e-02	1.50e-01	1.29e-05	3.64e-06
shaw	1000	1000	9	1.40e-02	1.74e-02	1.80e-04	1.58e-07
	1000	1000	13	3.47e-02	1.08e-01	1.42e-07	2.58e-09
	1000	1000	17	7.89e+00	1.62e+01	9.29e-07	1.47e-07
gravity	1000	1000	15	7.26e+02	2.11e+03	9.79e-04	1.29e-04
	1000	1000	19	1.21e+06	3.09e+06	5.73e-05	5.01e-06
	1000	1000	23	1.34e+04	3.89e+04	4.61e-06	1.58e-06
wing	1000	1000	3	3.74e-04	4.07e-04	6.09e-05	1.69e-07
	1000	1000	7	1.65e-01	3.88e-01	3.12e-08	9.81e-09
	1000	1000	11	6.42e+00	1.36e+01	-	-
foxgood	1000	1000	4	6.17e-02	1.45e-01	8.62e-04	3.20e-06
	1000	1000	8	7.88e-04	6.21e-04	1.48e-05	4.22e-07
	1000	1000	12	5.76e-01	7.46e-01	4.25e-07	2.59e-08

Table 2: CUR approximation of benchmark input matrices

7.3 CUR approximation with quasi-Gaussian preconditioning

Next we present the test results for CUR algorithms (Method 1 and 4) applied to hard input matrices for computational problems of two kinds, both replicated from [HMT11], namely, the matrices of

- (i) the discretized single-layer Laplacian operator and
- (ii) the approximation of the inverse of a finite-difference operator.

Direct application of CUR algorithm tended to produce results with large error. Thus we preconditioned the input matrix by multiplying it by 20 random bidiagonal matrices of (25), with random column permutations. For each setting we applied 10 different random multipliers, and then again we display the mean value and the standard deviation of the results.

Input matrices (i). We considered the Laplacian operator $[S\sigma](x) =$

$c \int_{\Gamma_1} \log |x - y| \sigma(y) dy, x \in \Gamma_2$, from [HMT11, Section 7.1], for two contours $\Gamma_1 = C(0, 1)$ and $\Gamma_2 = C(0, 2)$ on the complex plane. Its discretization defines an $n \times n$ matrix $M = (m_{ij})_{i,j=1}^n$ where $m_{i,j} = c \int_{\Gamma_{1,j}} \log |2\omega^i - y| dy$ for a constant c such that $\|M\| = 1$ and for the arc $\Gamma_{1,j}$ of the contour Γ_1 defined by the angles in $[\frac{2j\pi}{n}, \frac{2(j+1)\pi}{n}]$.

		Method 1		Method 4	
n	r	mean	std	mean	std
256	31	1.37e-04	2.43e-04	9.46e-05	2.11e-04
256	35	5.45e-05	7.11e-05	1.03e-05	1.08e-05
256	39	6.18e-06	6.32e-06	1.24e-06	1.72e-06
512	31	7.80e-05	6.00e-05	2.04e-05	1.52e-05
512	35	1.56e-04	1.53e-04	6.74e-05	1.79e-04
512	39	5.91e-05	1.10e-04	4.27e-05	1.20e-04
1024	31	9.91e-05	6.69e-05	2.79e-05	3.13e-05
1024	35	4.87e-05	4.35e-05	1.66e-05	1.50e-05
1024	39	6.11e-05	1.33e-04	3.83e-06	5.77e-06

Table 3: CUR Approximation of Laplacian input matrices

Inputs matrices (ii). We similarly applied our CUR algorithms (Method 1, 3, and 4) to the input matrix M being the inverse of a large sparse matrix representing a finite-difference operator from [HMT11, Section 7.2] and observed similar results with all structured and Gaussian multipliers.

		Method 1		Method 3		Method 4	
n	r	mean	std	mean	std	mean	std
800	78	4.85e-03	4.25e-03	3.30e-03	8.95e-03	3.71e-05	3.27e-05
800	82	2.67e-03	3.08e-03	4.62e-04	6.12e-04	2.23e-05	2.24e-05
800	86	2.14e-03	1.29e-03	4.13e-04	8.45e-04	6.73e-05	9.37e-05
1600	111	1.66e-01	4.71e-01	1.11e-03	1.96e-03	1.21e-04	1.17e-04
1600	115	3.75e-03	3.18e-03	1.96e-03	3.93e-03	4.03e-05	2.79e-05
1600	119	3.54e-03	2.27e-03	5.56e-04	7.65e-04	5.38e-05	8.49e-05
3200	152	1.87e-03	1.37e-03	3.23e-03	3.12e-03	1.68e-04	2.30e-04
3200	156	1.92e-03	8.61e-04	1.66e-03	1.65e-03	1.86e-04	1.17e-04
3200	160	2.43e-03	2.00e-03	1.98e-03	3.32e-03	1.35e-04	1.57e-04

Table 4: CUR approximation of finite difference matrices

We also performed tests on CGR with Abridged Fourier/Hadamard pre-processors (see [PZ16] Arxiv version for definitions). For aforementioned input matrices, Method 1 no longer provide stable results, while Method 2 still produce low-rank decompositions with small error. Table 5 displays the test results. (10 runs for each case)

7.4 Randomized factorization of Gaussian matrices

We have tested multiplication of 20 inverse-bidiagonal matrices (with random column permutation). We present the distribution of a single randomly chosen entry and the scattered

				Abridged Hadamard		Abridged Fourier	
Input Matrix	m	n	r	mean	std	mean	std
baart	1000	1000	4	1.14e-03	2.80e-06	1.36e-03	5.39e-06
shaw	1000	1000	9	4.63e-04	3.08e-05	5.78e-04	4.63e-05
gravity	1000	1000	16	1.98e-03	1.55e-04	1.70e-03	1.82e-04
wing	1000	1000	3	1.36e-04	2.51e-07	1.65e-04	2.14e-05
foxgood	1000	1000	4	1.62e-03	2.63e-04	1.97e-03	1.14e-04
Laplacian	256	256	15	4.08e-03	1.14e-03	3.94e-03	5.21e-04
	512	512	15	3.77e-03	1.34e-03	4.28e-03	6.07e-04
	1024	1024	15	3.97e-03	1.22e-03	4.09e-03	4.47e-04
finite difference	408	800	41	4.50e-03	1.12e-03	3.76e-03	8.36e-04
	808	1600	59	4.01e-03	1.10e-03	3.80e-03	1.70e-03
	1608	3200	80	4.60e-03	1.53e-03	3.85e-03	1.27e-03

Table 5: CGR Approximation (Method 2) with ASPF/ASPH pre-processors

plot of two such entries. The output 1024-by-1024 matrices were very close to Gaussian distribution and have passed the Kolmogorov-Smirnov test for normality in all tests repeated 1000 times.

8 Conclusions

We dramatically accelerated the known algorithms for the fundamental problems of CUR and low-rank approximation in the case of the average input matrix and then extended the resulting fast CUR algorithm to a wider class of inputs by means of multiplicative preprocessing.

Our extensive tests for a variety of real world inputs from a Singular Matrix Database have consistently supported the results of our formal analysis.

There are various challenging directions for further progress, e.g., empirical choice of more efficient preprocessing. Moreover, our accelerated CUR and low-rank approximation enables faster solution of some new important computational problems, thus extending the long list of the known applications (see [HMT11], [M11], [W14], [BW14], and the bibliography therein). In this section we are going to add two new highly important subjects to this long list.

8.1 Acceleration of the Fast Multipole Method (FMM) for the average HSS matrix

In this subsection we extend our fast CUR and low-rank approximation to the acceleration of a fundamental stage of the FMM, that is, multiplication by a vector of a HSS matrix⁸ defined by its average case generators.

We first recall that HSS matrices naturally extend the class of banded matrices and their inverses, are closely linked to FMM, have been intensively studied for decades (cf. [CGR88], [GR87], [T00], [BGH03], [GH03], [VVG05], [VVM07/08], [B10], [X12], [XXG12], [EGH13], [X13], [XXCB14], and the bibliography therein), and are highly and increasingly popular.

⁸Here and hereafter “HSS” stands for “hierarchically semiseparable”.

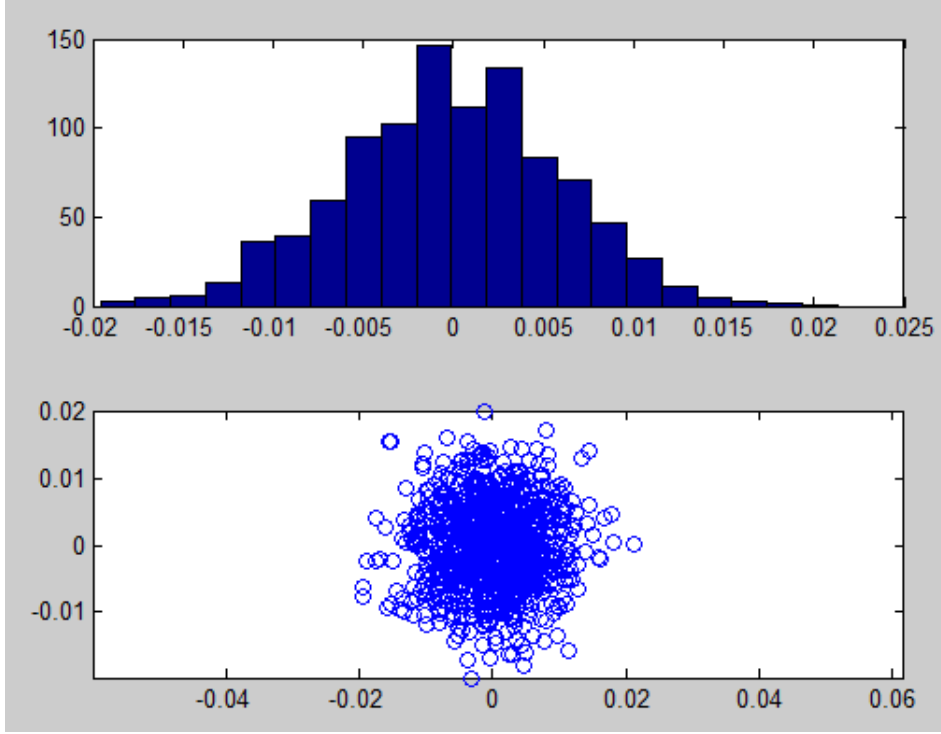


Figure 1: Distribution of a randomly chosen entry

Definition 36. (Cf. [MRT05].) With each diagonal block of a block matrix associate its complement in its block column, that is, the union of the pair of the maximal sub- and super-diagonal blocks in that block column, and call this complement a *neutered block column*.

Definition 37. (Cf. [X12], [X13], [XXCB14].)

A block matrix M of size $m \times n$ is called a r -HSS matrix, for a positive integer r ,

- (i) if all diagonal blocks of this matrix consist of $O((m+n)r)$ entries overall and
- (ii) if r is the maximum rank of its neutered block columns.

Remark 38. Many authors work with (l, u) -HSS (rather than r -HSS) matrices M for which l and u are the maximum ranks of the sub- and super-diagonal blocks, respectively. The (l, u) -HSS and r -HSS matrices are closely related. If a neutered block column N is the union of a sub-diagonal block B_- and a super-diagonal block B_+ , then $\text{rank}(N) \leq \text{rank}(B_-) + \text{rank}(B_+)$, and so an (l, u) -HSS matrix is a r -HSS matrix, for $r \leq l + u$, while clearly a r -HSS matrix is a (r, r) -HSS matrix.

The FMM exploits the r -HSS structure of a matrix as follows (cf. [VVM07/08], [B10], [EGH13]):

- (i) Cover all off-block-diagonal entries with a set of non-overlapping neutered block columns.
- (ii) Express every neutered block column N of this set as the product FH of two generator matrices, F of size $h \times r$ and H of size $r \times k$. Call the pair $\{F, H\}$ a *length r generator* of the neutered block column N .
- (iii) Multiply readily the matrix M by a vector by separately multiplying generators and diagonal blocks by subvectors, which involves $O((m+n)r)$ flops overall, and

(iv) in a more advanced application of FMM solve a nonsingular r -HSS linear system of n equations by using $O(nr \log^2(n))$ flops under some mild additional assumptions on the input.

This approach is readily extended to the same operations with (r, ξ) -HSS matrices, that is, matrices approximated by r -HSS matrices within a perturbation norm bound ξ where a positive tolerance ξ is small in context (e.g., is the unit round-off). Likewise, one defines an (r, ξ) -HSS representation and (r, ξ) -generators.

(r, ξ) -HSS matrices (for r small in context) appear routinely in matrix computations, and computations with such matrices are performed efficiently by using the above techniques.

In many applications of the FMM (cf., e.g., [BGP05], [VVVF10]) stage (ii) is omitted because short generators for all neutered block columns are readily available, but this is not the case in other important applications (cf. [XXG12], [XXCB14], and [P15]). The computation of such generators is precisely the low-rank approximation of the neutered block columns, which turns out to be the bottleneck stage of FMM in these applications.

Indeed apply random sampling Algorithm 6 at this stage with Gaussian multipliers. Multiplication of a $q \times h$ matrix by an $h \times r$ Gaussian matrix requires $(2h - 1)qr$ flops, while standard HSS-representation of an $n \times n$ HSS matrix includes $q \times h$ neutered block columns for $q \approx m/2$ and $h \approx n/2$. In this case the cost of computing an r -HSS representation of the matrix M is at least of order mnr . For large integers m and n , this greatly exceeds the above estimate of $O((m + n)r)$ flops at the other stages of the computations.

Alternative customary techniques for low-rank approximation rely on computing SVD or rank-revealing factorization of an input matrix and are at least as costly as the computations by means of random sampling.

Can we alleviate such a problem? Yes, for the average input we can compute low-rank approximations to (r, ξ) -generators much faster by applying our fast CUR approximation algorithms.

8.2 Acceleration of the Conjugate Gradient (CG) algorithms

We recall that a real $n \times n$ matrix M and a linear system of n equations $M\mathbf{x} = \mathbf{b}$ are said to be *symmetric positive definite*⁹ if $M = V^T V$ for a nonsingular matrix V [GL13] and use the following concept:

Definition 39. An $n \times n$ matrix M is (r, ξ) -concentrated if the set of its singular values is clustered (within a small tolerance ξ) about at most $r + 1$ values. Such a matrix is *strongly* (r, ξ) -concentrated if this set contains a cluster of at least $n - r$ singular values, each counted with its multiplicity.

The following two facts show efficiency of the CG algorithms:

(i) Given a spd linear system of equations $M\mathbf{x} = \mathbf{b}$ whose matrix M is (r, ξ) -concentrated, the CG algorithms converge to its solution within an error norm in $O(\xi)$ in at most r iterations [A94], [BBC93], [G97], [B02], [S03].

(ii) Various highly important present day computations routinely involve matrices made strongly (r, ξ) -concentrated and hence (r, ξ) -concentrated, for reasonably small integers r and small positive ξ , by means of some standard *preconditioning* techniques.

⁹Hereafter we use the acronym *spd*.

The next critical issue is whether we can decrease the computational cost of a CG iteration, which is reduced essentially to computing or closely approximating the product of the matrix M by a vector or by a few vectors.

Next we prove that strongly (r, ξ) -concentrated matrices are also (r, ξ) -HSS matrices, and so, by applying our accelerated variant of FMM, we can approximate the product of such an $n \times n$ average matrix M by a vector significantly faster than by applying the known algorithms, which involve $(2n - 1)n$ flops.

Theorem 40. *If an $n \times n$ spd matrix M is strongly r -concentrated, then the numerical rank of any of its off-diagonal submatrix is at most r .*

Proof. Since M is a strongly r -concentrated matrix, at least $n - r$ its singular values are clustered about a certain value s . Change all the singular values by assigning to them this value s and denote the resulting matrix $\widehat{M} = M + E$ where the matrix E has numerical rank at most r and where $\widehat{M} = U\widehat{S}U^T$, for $\widehat{S} = sI_n$ and an orthogonal matrix U . Note that in this case $\widehat{M} = sI_n$, and so every off-diagonal submatrix of \widehat{M} is filled with 0s. Therefore the matrices M and E share all their off-diagonal submatrices. Consequently numerical rank of such a submatrix cannot exceed $\text{nrnk}(E) \leq r$. \square

Corollary 41. If an $n \times n$ spd matrix M is strongly r -concentrated, then we can approximate the solution of a linear system of n equations $M\mathbf{x} = \mathbf{b}$ by using $O(r^2 n \log(n)) + \gamma(M)$ flops provided that one can compute generators of length at most r for a r -HSS approximate representation of the matrix M by using $\gamma(M)$ flops.

By virtue of Theorem 40 the neutered block columns in the r -HSS representation of the matrix M have numerical ranks at most r . By virtue of our results of the previous subsection, $\gamma(M) = O(rn)$ in the case of the average matrix M , implying respective acceleration of the CG algorithms.

Remark 42. Extension to nonsymmetric inputs. Recall that any nonsingular linear system $V\mathbf{x} = \mathbf{b}$ is equivalent to the spd linear systems $V^T V\mathbf{x} = \mathbf{c}$ and $VV^T \mathbf{y} = \mathbf{b}$ for $\mathbf{c} = V^T \mathbf{b}$ and $\mathbf{x} = V^T \mathbf{y}$. Therefore we can extend our results to a nonsymmetric nonsingular linear system of equations $A\mathbf{y} = \mathbf{f}$ by means of symmetrization of a matrix A in any of the two ways, $A \rightarrow M = A^T A$ or $A \rightarrow M = AA^T$, and then application of the CG algorithms to the matrix M defined implicitly by the above products, and never computed explicitly. This leads to the *CG normal equation error* method and the *CG normal equation error* method, respectively (cf. [GL13, Section 11.3.9]), to which we can extend our study of the CG method.

Appendix

A Randomized Matrix Computations

Definition 43. (i) Call an $m \times n$ matrix *Gaussian* if all its entries are i.i.d. Gaussian variables. $\mathcal{G}^{m \times n}$ denotes the class of $m \times n$ Gaussian matrices.

(ii) If $G \in \mathcal{G}^{m \times r}$, $H \in \mathcal{G}^{r \times n}$, $\Sigma = \text{diag}(\sigma_i)_{i=1}^r$, $\sigma_1 = 1 \geq \sigma_2 \geq \dots, \sigma_r > 0$, and the value $\eta_+ = 1/\sigma_r$ is not large, then call the matrix $W := G\Sigma H$ a *scaled factor-Gaussian matrix of expected rank r* and write $W \in \mathcal{G}_{m,n,r}(\Sigma)$.

(iii) In the special case where $\Sigma = I_r$, $W \in \mathcal{G}_{m,n,r}(I_r)$, call an $m \times n$ matrix $W = G\Sigma H = GH$ a *factor-Gaussian matrix of expected rank r* and write $W \in \mathcal{G}_{m,n,r}$.

(iv) Call an $m \times n$ matrix $W = GH$ an l -factor-Gaussian matrix of expected rank r provided that $G \in \mathcal{G}^{m \times r}$, $H \in \mathbb{C}^{r \times n}$, $\|H\| = 1$, and the value $h_+ = \|H^+\|$ is not large.

(v) Call a matrix $W = GH$ a r -factor-Gaussian matrix of expected rank r provided that $H \in \mathcal{G}^{r \times n}$, $\|G\| = 1$, and the value $g_+ = \|G^+\|$ is not large.¹⁰

Theorem 44. Suppose that A is an $m \times n$ matrix of full rank $k = \min\{m, n\}$, F and H are $r \times m$ and $n \times r$ matrices, respectively, for $r \leq k$, and the entries of these two matrices are nonconstant linear combinations of finitely many i.i.d. random variables v_1, \dots, v_h .

Then the matrices F , FA , H , and AH have full rank r

(i) with probability 1 if v_1, \dots, v_h are Gaussian variables and

(ii) with a probability at least $1 - r/|\mathcal{S}|$ if they are random variables sampled under the uniform probability distribution from a finite set \mathcal{S} having cardinality $|\mathcal{S}|$.

Proof. The determinant, $\det(B)$, of any $r \times r$ block B of a matrix F , FA , H , or AH is a polynomial of degree r in the variables v_1, \dots, v_h , and so the equation $\det(B) = 0$ defines an algebraic variety of a lower dimension in the linear space of these variables (cf. [BV88, Proposition 1]). Clearly such a variety has Lebesgue and Gaussian measures 0, both being absolutely continuous with respect to one another. This implies claim (i) of the theorem. Derivation of claim (ii) from a celebrated lemma of [DL78], also known from [Z79] and [S80], is a well-known pattern, specified in some detail in [PW08]. \square

Remark 45. The theorem implies that the matrices G , H and W of Definition 43 have rank r with probability 1.

We state the following estimates for real Gaussian matrices, but similar estimates in the case of complex matrices can be found in [D88], [E88], [E89], [CD05], and [ES05].

Definition 46. Norms of random matrices and the expected value of a random variable. Write $\nu_{m,n} := \|G\|$, $\nu_{m,n}^+ := \|G^+\|$, $\nu_{m,n,C} := \|G\|_C$, and $\nu_{m,n,F}^+ := \|G^+\|_F$, for a Gaussian $m \times n$ matrix G , and write $\mathbb{E}(v)$ for the expected value of a random variable v . ($\nu_{m,n} = \nu_{n,m}$, $\nu_{m,n}^+ = \nu_{n,m}^+$, and $\nu_{F,m,n} = \nu_{F,n,m}$, for all pairs of m and n .)

Theorem 47. Let m and n be positive integers and let $t \geq 0$. Then

- (i) Probability $\{\nu_{m,n} > t + \sqrt{m} + \sqrt{n}\} \leq \exp(-t^2/2)$ (see [DS01, Theorem II.7]),
- (ii) $\mathbb{E}(\nu_{m,n}) \leq \sqrt{m} + \sqrt{n}$ (see [HMT11, Proposition 10.1] for $S = T = I$), and
- (iii) $\mathbb{E}(\nu_{m,n,C}) = \mathbb{E}(\nu_{1,mn,C}) \leq f + \frac{1}{f}$, for $f = \sqrt{2 \ln(\max\{2, mn\})}$ (see [SST06, Lemma A.3]).

Remark 48. Given an $m \times n$ Gaussian matrix W , apply Theorem 47 and obtain that $E(\|W\|) \leq \sqrt{m} + \sqrt{n}$, $E(\|W\|_C) \leq f + \frac{1}{f}$ where $f \approx \sqrt{2 \ln(mn)}$ for large integers mn ; furthermore actually both norms $\|W\|$ and $\|W\|_C$ deviate from their expected values by more than a factor $\zeta > 1$ with a probability that decays exponentially fast as ζ grows to infinity. These observations imply a tighter expected range for the ratio of the spectral and Chebyshev norms of a Gaussian matrix versus bounds (2) for general matrix.

Theorem 49. Let $\Gamma(x) := \int_0^\infty \exp(-t)t^{x-1}dt$ denote the Gamma function and let $x > 0$. Then

- (i) Probability $\{\nu_{m,n}^+ \geq mx^2\} < \frac{x^{n-m-1}}{\Gamma(m-n+2)}$ for $m \geq n \geq 2$,
- (ii) Probability $\{\nu_{n,n}^+ \geq x\} \leq 2.35\sqrt{n}/x$ for $n \geq 2$,
- (iii) $\mathbb{E}(\nu_{m,n}^+) \leq e\sqrt{m}/|m-n|$, provided that $m \neq n$ and $e = 2.71828\dots$

¹⁰In parts (iv) and (v) “l” and “r” are the acronyms for “left” and “right”.

Proof. See [CD05, Proof of Lemma 4.1] for claim (i), [SST06, Theorem 3.3] for claim (ii), and [HMT11, Proposition 10.2] for claim (iii). \square

The probabilistic upper bounds of Theorem 49 on $\nu_{m,n}^+$ are quite reasonable even where $m = n$, but are strengthened very fast as the difference $|m - n|$ grows from 1. We also recall that with a probability close to 1, the smallest singular value of a $r \times n$ Gaussian matrix is not less than $\sqrt{n} - \sqrt{r}$ (cf. [RV09], [TV10]).

Remark 50. Theorems 47 and 49 combined imply that an $m \times n$ Gaussian matrix is well-conditioned unless the integer $m + n$ is large or the integer $|m - n|$ is close to 0. With some grain of salt we can consider such a matrix well-conditioned even where the integer $|m - n|$ is small or vanishes. Clearly, these properties hold for all submatrices of a Gaussian matrix as well.

B Volume maximization

B.1 Some known algorithms

Theorems 16 and 17 motivate maximization of the volume or the r -projective volume of $k \times l$ submatrices $W_{\mathcal{I}_q, \mathcal{J}_s}$ at stages 1 and 2 of cross-approximation Algorithm 12.

Unfortunately, however, we have no low-cost algorithm that would maximize the volume or the r -projective volume for the worst case input of stages 1 and 2, but we can alternate application of the known heuristic algorithms in various iteration loops of Algorithm 12, at stages 1 and 2 of the same loop, and possibly even within the same stage.

Given an $m \times l$ matrix W and a positive integer r , $1 \leq r \leq l \leq m$, the algorithms of the papers [GE96] and [P00] perform $O(ml^2)$ flops in order to compute its $r \times l$ submatrix $W_{\mathcal{I}, \mathcal{J}}$ having locally maximal volume. For a $k \times n$ matrix these algorithm use $O(nk^2)$ flops and compute a $k \times r$ submatrix having locally maximal volume.

We have no *á priori* estimates for the ratio $h = v_{2, \max}/v_2(W_{\mathcal{I}, \mathcal{J}})$ where $v_{2, \max}$ denotes the maximal volume of the submatrices of W having the same size and submatrix $W_{\mathcal{I}, \mathcal{J}}$ has locally maximal volume, but we can obtain such estimates *á posteriori* (see the next subsection), and we have the following result, which is attractive in view of Corollary 4 and Theorem 27.

Theorem 51. (See [GZT95], [GTZ97], [GTZ97a], [GE96], [P00, Lemma 3.5].) For an $m \times n$ matrix W and its $k \times l$ submatrix $W_{\mathcal{I}, \mathcal{J}}$, it holds that

$$\sqrt{(n-r)r h_c^2 + 1} \sigma_r(W_{\mathcal{I}, \mathcal{J}}) \geq \sigma_r(W_{\mathcal{I}, :})$$

if $k = r \leq l$ and if the volume $v_2(W_{\mathcal{I}, \mathcal{J}})$ is locally column-wise h_c -maximal and

$$\sqrt{(m-r)r h_r^2 + 1} \sigma_r(W_{\mathcal{I}, \mathcal{J}}) \geq \sigma_r(W_{:, \mathcal{J}})$$

if $k \geq l = r$ and if this volume is locally row-wise h_r -maximal.

Proof. The theorem turns into [P00, Lemma 3.5] for $k = l = r$ and is extended to the case where $r = \min\{k, l\}$ because a singular value of a $k \times l$ matrix does not increase in the transition to its $r \times r$ submatrix. \square

Clearly, the theorem applies to the matrices $W_{\mathcal{I},\mathcal{J}}$ maximizing volumes globally as well because such volumes are also locally maximal.

The iterative algorithm of [GOSTZ10] seeks a $r \times r$ submatrix of locally maximal volume in a $r \times n$ matrix. The algorithm is initiated with $O(r^3)$ flops, then performs every iteration by using $O(nr)$ flops, and strictly increases volume at every iteration, has no good upper bounds on the number of iterations, but empirically is quite efficient.

B.2 Some simple estimates for the maximal volume

One can apply the following upper bounds on the volume of matrices.

(i) For a $r \times r$ matrix $M = (m_{ij})_{i,j=1}^r = (\mathbf{m}_j)_{j=1}^r = (\bar{\mathbf{m}}_i^*)_{i=1}^r$ where $\mathbf{m}_j = (m_{ij})_{i=1}^r$ and $\bar{\mathbf{m}}_i = ((m_{ij})_{j=1}^r)^*$ for all i and j , recall the *Hadamard's bounds*

$$v_2(M) = |\det(M)| \leq \min \left\{ \prod_{j=1}^r \|\mathbf{m}_j\|, \prod_{i=1}^r \|\bar{\mathbf{m}}_i^*\|, r^{r/2} \max_{i,j=1}^r |m_{ij}|^r \right\}.$$

For a $k \times l$ matrix M and $k \leq l$, it holds that $v_2(M) = \det(M^*M) = \det(\mathbf{m}_i^* \mathbf{m}_j)_{i,j=1}^k$, and so

$$v_2(M)^2 = v_2(M^*)^2 \leq \min \left\{ \prod_{j=1}^k \sum_{i=1}^k (\mathbf{m}_i^* \mathbf{m}_j)^2, \prod_{i=1}^k \sum_{j=1}^k (\mathbf{m}_i^* \mathbf{m}_j)^2, k^k \max_{i,j=1}^k |\mathbf{m}_i^* \mathbf{m}_j|^{2k} \right\}.$$

(ii) For a $k \times q$ matrix M and any of its $k \times l$ submatrix $M_{k,l}$ where $k \leq l \leq q$, it holds that

$$v_2(M) = v_{2,k}(M) \geq v_2(M_{k,l}) = v_{2,k}(M_{k,l}).$$

Acknowledgements: Our research has been supported by NSF Grants CCF-1116736 and CCF-1563942 and PSC CUNY Award 68862-00 46. We are also grateful to S. A. Goreinov, E. E. Tyrtshnikov, and P. Chakraborty for helpful pointers to the bibliography and to A. Osinsky and N. L. Zamarashkin for a preprint of [O16].

References

- [A94] O. Axelsson, *Iterative Solution Methods*, Cambridge Univ. Press, Cambridge, England, 1994.
- [AW07] A. C. Elliott, W. A. Woodward, *Statistical analysis quick reference guidebook: With SPSS examples*, Sage, 2007.
- [B41] A. C. Berry, The accuracy of the Gaussian approximation to the sum of independent variates, *Transactions of the American Mathematical Society*, **49**, **1**, 122–136, 1941.
- [B02] M. Benzi, Preconditioning Techniques for Large Linear Systems: a Survey, *J. of Computational Physics*, **182**, 418–477, 2002.
- [B10] S. Börm, *Efficient Numerical Methods for Non-local Operators: \mathcal{H}^2 -Matrix Compression, Algorithms and Analysis*, European Math. Society, 2010.

- [BBC93] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1993.
- [BGH03] S. Börm, L. Grasedyck, W. Hackbusch, Introduction to Hierarchical Matrices with Applications, *Engineering Analysis with Boundary Elements*, **27**, **5**, 405–422, 2003.
- [BGP05] A. Bini, L. Gemignani, V. Y. Pan, Fast and Stable QR Eigenvalue Algorithms for Generalized Semiseparable Matrices and Secular Equation, *Numerische Mathematik*, **100**, **3**, 373–408, 2005.
- [B-I92] A. Ben-Israel, A volume associated with $m \times n$ matrices, *Linear Algebra and its Applications*, **167**, 87–111, 1992.
- [BV88] W. Bruns, U. Vetter, *Determinantal Rings, Lecture Notes in Math.*, **1327**, Springer, Heidelberg, 1988.
- [BW14] C. Boutsidis, D. Woodruff, Optimal CUR Matrix Decompositions, *ACM Symposium on Theory of Computing (STOC 2014)*, 353–362, ACM Press, New York, 2014. Full version in arXiv:1405.7910v2 [cs.DS] 16 July 2014.
- [BY13] L. A. Barba, R. Yokota, How Will the Fast Multipole Method Fare in Exascale Era? *SIAM News*, **46**, **6**, 1–3, July/August 2013.
- [C46] Harald Cramér, *Mathematical Methods of Statistics*, Princeton University Press, 575 pages, 1999 (first edition 1946), ISBN: 9780691005478.
- [C00] Barry A Cipra, The Best of the 20th Century: Editors Name Top 10 Algorithms, *SIAM News*, **33** **4**, 2, published by the Society for Industrial and Applied Mathematics, May 16, 2000.
- [CD05] Z. Chen, J. J. Dongarra, Condition Numbers of Gaussian Random Matrices, *SIAM J. on Matrix Analysis and Applications*, **27**, 603–620, 2005.
- [CGR88] J. Carrier, L. Greengard, V. Rokhlin, A Fast Adaptive Algorithm for Particle Simulation, *SIAM Journal on Scientific Computing*, **9**, 669–686, 1988.
- [CI94] S. Chandrasekaran, I. Ipsen, On rank revealing QR factorizations, *SIAM J. Matrix Anal. Appl.*, **15**, 592–622, 1994.
- [D88] J. Demmel, The Probability That a Numerical Analysis Problem Is Difficult, *Math. of Computation*, **50**, 449–480, 1988.
- [DL78] R. A. Demillo, R. J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Processing Letters*, **7**, **4**, 193–195, 1978.
- [DS01] K. R. Davidson, S. J. Szarek, Local Operator Theory, Random Matrices, and Banach Spaces, in *Handbook on the Geometry of Banach Spaces* (W. B. Johnson and J. Lindenstrauss editors), pages 317–368, North Holland, Amsterdam, 2001.
- [DZBLCF14] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation, *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS’14)*, 1269–1277, MIT Press, Cambridge, MA, USA, 2014.

- [E88] A. Edelman, Eigenvalues and Condition Numbers of Random Matrices, *SIAM J. on Matrix Analysis and Applications*, **9**, **4**, 543–560, 1988.
- [E89] A. Edelman, Eigenvalues and condition numbers of random matrices, Ph.D. thesis, Massachusetts Institute of Technology, 1989.
- [EGH13] Y. Eidelman, I. Gohberg, I. Haimovici, *Separable Type Representations of Matrices and Fast Algorithms, volumes 1 and 2*, Birkhäuser, 2013.
- [ES05] A. Edelman, B. D. Sutton, Tails of Condition Number Distributions, *SIAM J. on Matrix Analysis and Applications*, **27**, **2**, 547–560, 2005.
- [G97] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
- [GE96] M. Gu, S.C. Eisenstat, An efficient algorithm for computing a strong rank revealing QR factorization, *SIAM J. Sci. Comput.*, **17**, 848–869, 1996.
- [GH03] L. Grasedyck, W. Hackbusch, Construction and Arithmetics of H-Matrices, *Computing*, **70**, **4**, 295–334, 2003.
- [GL13] G. H. Golub, C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 2013 (fourth edition).
- [GOSTZ10] S. Goreinov, I. Oseledets, D. Savostyanov, E. Tyrtyshnikov, and N. Zamarashkin, How to find a good submatrix, in *Matrix Methods: Theory, Algorithms, Applications*, (dedicated to the Memory of Gene Golub, edited by V. Olshevsky and E. Tyrtyshnikov), pages 247–256, World Scientific Publishing, New Jersey, ISBN-13 978-981-283-601-4, ISBN-10-981-283-601-2 2010.
- [GR87] L. Greengard, V. Rokhlin, A Fast Algorithm for Particle Simulation, *Journal of Computational Physics*, **73**, 325–348, 1987.
- [GT01] S. A. Goreinov, E. E. Tyrtyshnikov, The maximal-volume concept in approximation by low-rank matrices, *Contemporary Mathematics*, **208**, 47–51, 2001.
- [GT11] S. A. Goreinov, E. E. Tyrtyshnikov, Quasioptimality of skeleton approximation of a matrix on the Chebyshev norm, *Russian Academy of Sciences: Doklady*, **83**, **3**, 1–2, 2011.
- [GTZ97] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A Theory of Pseudo-skeleton Approximations, *Linear Algebra and Its Applications*, **261**, 1–21, 1997.
- [GTZ97a] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, Pseudo-skeleton Approximations by Matrices of Maximal Volume, *Mathematical Notes*, **62**, **4**, 515–519, 1997.
- [GZT95] S. A. Goreinov, N. L. Zamarashkin, E. E. Tyrtyshnikov, Pseudo-skeleton Approximations of Matrices, *Russian Academy of Sciences: Doklady*, **343**, **2**, 151–152, 1995.
- [HC94] Hansen, Per Christian. "Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems." *Numerical algorithms* 6.1 (1994): 1-35.

- [HMT11] N. Halko, P. G. Martinsson, J. A. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, *SIAM Review*, **53**, 2, 217–288, 2011.
- [JVZ14] M. Jaderberg, A. Vedaldi, A. Zisserman, Speeding up Convolutional Neural Networks with Low Rank Expansions, arXiv:1405.3866 CS, May 2014, and *BMVC* 2014.
- [M11] M. W. Mahoney, Randomized Algorithms for Matrices and Data, *Foundations and Trends in Machine Learning*, NOW Publishers, **3**, 2, 2011. (Abridged version in: *Advances in Machine Learning and Data Mining for Astronomy*, edited by M. J. Way, et al., pp. 647-672, 2012.)
- [M14] A. Ju. Mikhalev, Method for the Low-rank Block Approximation of a Matrix from Its Entries, PhD Thesis, 105 pages, 2014.
- [MRT05] P. G. Martinsson, V. Rokhlin, M. Tygert, A Fast Algorithm for the Inversion of General Toeplitz Matrices, *Comput. Math. Appl.*, **50**, 741–752, 2005.
- [O16] A.I. Osinsky, Error Estimates for Pseudo-skeleton Approximations of Matrices, preprint, 2016.
- [P00] C.-T. Pan, On the existence and computation of rank-revealing LU factorizations, *Linear Algebra and its Applications*, **316**, 199–222, 2000.
- [P15] V. Y. Pan, Transformations of Matrix Structures Work Again, *Linear Algebra and Its Applications*, **465**, 1–32, 2015.
- [PW08] V. Y. Pan, X. Wang, Degeneration of Integer Matrices Modulo an Integer, *Linear Algebra and Its Applications*, **429**, 2113–2130, 2008.
- [PZ16] V. Y. Pan, L. Zhao, Low-rank Approximation of a Matrix: Novel Insights, New Progress, and Extensions, Proc. of the *Eleventh International Computer Science Symposium in Russia (CSR'2016)*, (Alexander Kulikov and Gerhard Woeginger, editors), St. Petersburg, Russia, June 2016, *Lecture Notes in Computer Science (LNCS)*, **9691**, 352–366, Springer International Publishing, Switzerland (2016). Also arXiv:1510.06142 [math.NA], submitted on 21 Oct 2015, revised June 2016.
- [PZ17] V. Y. Pan, L. Zhao, New Studies of Randomized Augmentation and Additive Pre-processing, *Linear Algebra Appl.*, 2017, <http://dx.doi.org/10.1016/j.laa.2016.09.035>
- [PZa] V. Y. Pan, L. Zhao, Numerically Safe Gaussian Elimination with No Pivoting, arxiv 1501.05385 CS, submitted on January 22, 2015, revised in June 2016.
- [RV09] M. Rudelson, R. Vershynin, The smallest singular value of a rectangular random matrix, *Comm. Pure Appl. Math.*, **62**, 12, 1707–1739, 2009.
- [S80] J. T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of ACM*, **27**, 4, 701–717, 1980.
- [S98] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decompositions*, SIAM, Philadelphia, 1998.

- [S03] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
- [SKSAR13] T. N. Sainath, B. Kingsbury, V. Sindhvani, E. Arisoy, B. Ramabhadran, Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets, *Proceedings of the 38th IEEE International Conference Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, 6655–6659, IEEE Computer Society Press, 2013. Doi: 10.1109/ICASSP.2013.6638949
- [SL14] A. Senior, X. Lei, Fine context, low-rank, softplus deep neural networks for mobile speech recognition, *Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, 7644–7648, IEEE Computer Society Press, 2014.
- [SMH07] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann Machines for Collaborative Filtering, *Proceedings of the 24th International Conference on Machine learning (ICML '07)*, 791–798, ACM Press, New York, NY, USA, 2007. Doi: 10.1145/1273496.1273596
- [SST06] A. Sankar, D. Spielman, S.-H. Teng, Smoothed Analysis of the Condition Numbers and Growth Factors of Matrices, *SIAM J. on Matrix Analysis and Appl.*, **28**, **2**, 446–476, 2006.
- [T96] E. Tyrtyshnikov, Mosaic-skeleton approximations, *Calcolo*, **33**, **1**, 47–57, 1996.
- [T00] E.E. Tyrtyshnikov, Incomplete Cross-Approximation in the Mosaic-Skeleton Method, *Computing*, **64**, 367–380, 2000.
- [TV10] Terence Tao, Van Vu, Random matrices: The distribution of the smallest singular values, *Geometric and Functional Analysis*, **20**, **1**, 260–297, 2010.
- [VVG05] R. Vandebril, M. Van Barel, G. Golub, N. Mastronardi, A Bibliography on Semiseparable Matrices, *Calcolo*, **42**, **3–4**, 249–270, 2005.
- [VVM07/08] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices* (Volumes 1 and 2), The Johns Hopkins University Press, Baltimore, Maryland, 2007.
- [VVVF10] M. Van Barel, R. Vandebril, P. Van Dooren, K. Frederix, Implicit Double Shift *QR*-algorithm for Companion Matrices, *Numerische Mathematik*, **116**, 177–212, 2010.
- [W14] D.P. Woodruff, Sketching As a Tool for Numerical Linear Algebra, *Foundations and Trends in Theoretical Computer Science*, **10**, **1–2**, 1–157, 2014.
- [X12] J. Xia, On the Complexity of Some Hierarchical Structured Matrix Algorithms, *SIAM J. Matrix Anal. Appl.*, **33**, 388–410, 2012.
- [X13] J. Xia, Randomized Sparse Direct Solvers, *SIAM J. Matrix Anal. Appl.*, **34**, 197–227, 2013.
- [XXCB14] J. Xia, Y. Xi, S. Cauley, V. Balakrishnan, Superfast and Stable Structured Solvers for Toeplitz Least Squares via Randomized Sampling, *SIAM J. Matrix Anal. Appl.*, **35**, 44–72, 2014.

- [XXG12] J. Xia, Y. Xi, M. Gu, A Superfast Structured Solver for Toeplitz Linear Systems via Randomized Sampling, *SIAM J. Matrix Anal. Appl.*, **33**, 837–858, 2012.
- [Z79] R. E. Zippel, Probabilistic Algorithms for Sparse Polynomials, *Proceedings of EU-ROSAM'79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.